Spring 4-1-2011

Health Club Management System

Nurunnahar Khanam
Dakota State University

HEALTH CLUB MANAGEMENT SYSTEM

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

April, 2011

By

Nurunnahar Khanam

Project Committee:

Dr. Ronghua Shan
Dr. Rick Christoph
Patti Brooks

DAKOTA STATE
dsu
UNIVERSITY

PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Nurunnahar Khanam

Master's Project Title: Health Club Management System

Faculty supervisor: Ronghua Shan Date: 3/30/2011

Committee member: Rick Christoph Date: 3/30/11

Committee member: Patti Brooks Date: 3-30-11

ABSTRACT

This project is about to automate and stream line the business processes of Bellevue Health Club Management System (BHC). The system will automate the daily tasks for managers and employees of the Health Club and allows best use of the resources. The system will provide a high level of service expected in the current competitive market. It will make business management system's daily operation faster and cut down on time wasted and make the company more competitive on the market. Various analysis methods and strategies have been applied during the development phase of the project. HCM used to use a manual process for tracking member information, employee information, etc. This process has been automated in the developed system by applying Process Reengineering mechanism. By taking this approach the company completely changes the way the business it performs. It changes the way of managing member information and offering a member self-service web-interface. As with all BPR-based projects, the risk of failure of this project was the major challenging factor. Adopted BPR analysis strategy helped to ensure the best way to meet the needs of the customers and the staffs of the company. This document describes the report on the developed system including system design, technical documentation, coding and database design.

Declaration

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

NKhanam
Nurunnahar Khanam

TABLE OF CONTENTS

Table of Figures

CHAPTER 1

INTRODUCTION

Background of the Problem:

The Bellevue Health Club (BHC) is health club that provides service among the local communities. The company is conducting business processes manually, which is very tedious and cumbersome. Currently, it does not have a web interface for customer interaction. Schedule personal trainer, member checking, member and trainer login process is also tracked manually. Generating report, billing, managing membership and communication with customer is becoming problematic. The demand for health club service is steadily increasing and the business process is slow and outmoded. In order to maintain the high level of service expected, BHC need to develop some stream lined methods to make its daily operation faster and efficient. The project completely differs from the existing system. One of the existing systems uses manual approach but the proposed project uses modern technology for conducting these operations. The proposed system will provide users a single interface to access customer information, view the status of the training, and schedule training, billing, and communication with customers. This project report explains the details of the problems that forced health club management department to develop this application, details of project management approach adopted, and the technological approach adopted to solve the problems. The report also explains the database designs of various modules and the usage of the application.

Company Background:

Bellevue Health Club is known for its friendly health club service among the local community. It has around thirty employees including managers, administrators, personal trainers and clerks. It offers various type of health related services including membership and training. The company is conducting all business processes manually, which is very tedious and cumbersome. Currently, it does not have a web interface for customer interaction. Schedule personal trainer, member checking, member and trainer login process is also tracked manually. To streamline the process and increase the revenue and volume of the business, the company is in need of an information system.

CHAPTER 2

PRBOBLEM STATEMENT

The management of the Health Club decided to automate their business processes by developing a system that will manage membership, schedule personal trainer, online scheduling and registration, reporting, billing, member checking, and member and trainer login processes. Admin will create account for every member, staff, and trainer, admin will also schedule personal trainer and training, generate report by using the new system. Member will login to the system to view the account and any updated information about the training and bill and report. They can also search available training session. Trainer can login to the system to view their account, report and bill statement, and training session. Current system do all these task manually, which very tough and time wastes. This process has been automated in the developed system by applying Process Reengineering mechanism. By taking this approach the company completely changes the way the business it performs. It changes the way of managing member information and offering a member self-service web-interface. As with all BPR-based projects, the risk of failure of this project was the major challenging factor. Adopted BPR analysis strategy helped to ensure the best way to meet the needs of the customers and the staffs of the company.

CHAPTER 3

OBJECTIVE OF THE PROJECT

Facing extreme dissatisfaction from the member of the health club and management department, manager took initiatives to develop a web-based system, which are aimed to help the member with their day-to-day operations. The planned solution has two different modules

1) Automated Business Process.
2) Web-Based Interface for the member.

The solution will include the following powerful features, which are identified as essential to meet the member requirements.

3.1 Essential features

- Automatic Business Process

- Functionality to automatically manage membership, schedule personal trainer, online scheduling and registration, reporting, billing, member checking, and member and trainer login processes.

- Simple and Flexible Front-end

- The application provides a simple and easy to use front-end in a web browser

- Scalable Application: As the demand for the membership of BHC grows in size the application can be scaled to fit the needs easily. Due to the loosely coupled nature of the tie between Application and Database each unit can be changed independent of another

- Easy to Maintain: The application will help to maintain tasks involved in database. This saves time and money for BHC.

- Higher availability and Faster Response Times.

The application is designed in such a way that it will be available to users 24/7. The application is also well tuned and provides better response time than the legacy applications used at BHC.

The following section is the work break down structure, which describes the various phases in the Development.

3.2 Work Breakdown Structure

3.2.1 Initiating

3.2.1.1 Identify the need for an automated system
3.2.1.2 Receive project sponsorship (project sponsor defined)
3.2.1.3 Assign the project manager
3.2.1.4 Identify the key stakeholders
3.2.1.5 Create the business case

3.2.1.6 The team contract

3.2.1.7 Create the project charter

3.2.2 Planning and Analysis

3.2.2.1 Meeting with key stakeholders

3.2.2.2 Identify risks

3.2.2.3 Create the project scope statement

3.2.2.4 Project cost and task breakdown

3.2.2.5 Determine the required resources

3.2.2.6 Assign specific tasks

3.2.2.7 Determine the task durations

3.2.2.8 Create the Work Breakdown Structure

3.2.2.9 Capture requirements for new system

3.2.2.10 Capture hardware requirements

3.2.2.11 Create the quality testing plan

3.2.2.12 Create the project communications plan

3.2.3 Execution

3.2.3.1 User interviews to obtain system requirements

3.2.3.2 System design

3.2.3.3 Develop logical design

3.2.3.4 Determine database architecture

3.2.3.5 Develop physical design

3.2.3.6 Design GUI mockups

3.2.3.7 Obtain user feedback on visual design

3.2.4 Develop System

3.2.4.1 Code appropriately

3.2.4.2 Create fully operational interactions

3.2.5 Testing

3.2.5.1 Testing of system, bugs fixed

3.2.6 Systems Implementation

3.2.6.1 Hardware installed

3.2.6.2 Server installed

3.2.6.3 Workstation installed

3.2.7.4 Software installed

3.2.8Training

3.2.8.1 Train users

3.2.9 Ongoing monitoring and maintenance

3.2.9.1 Annual Maintenance

- save time for customer and employee
- lower the work presser for employee
- increase the customer satisfaction
- increase the quality of service

3.3 Gantt Chart:

The Gantt chart provides the time line for the series of activities described in the work break down structure.

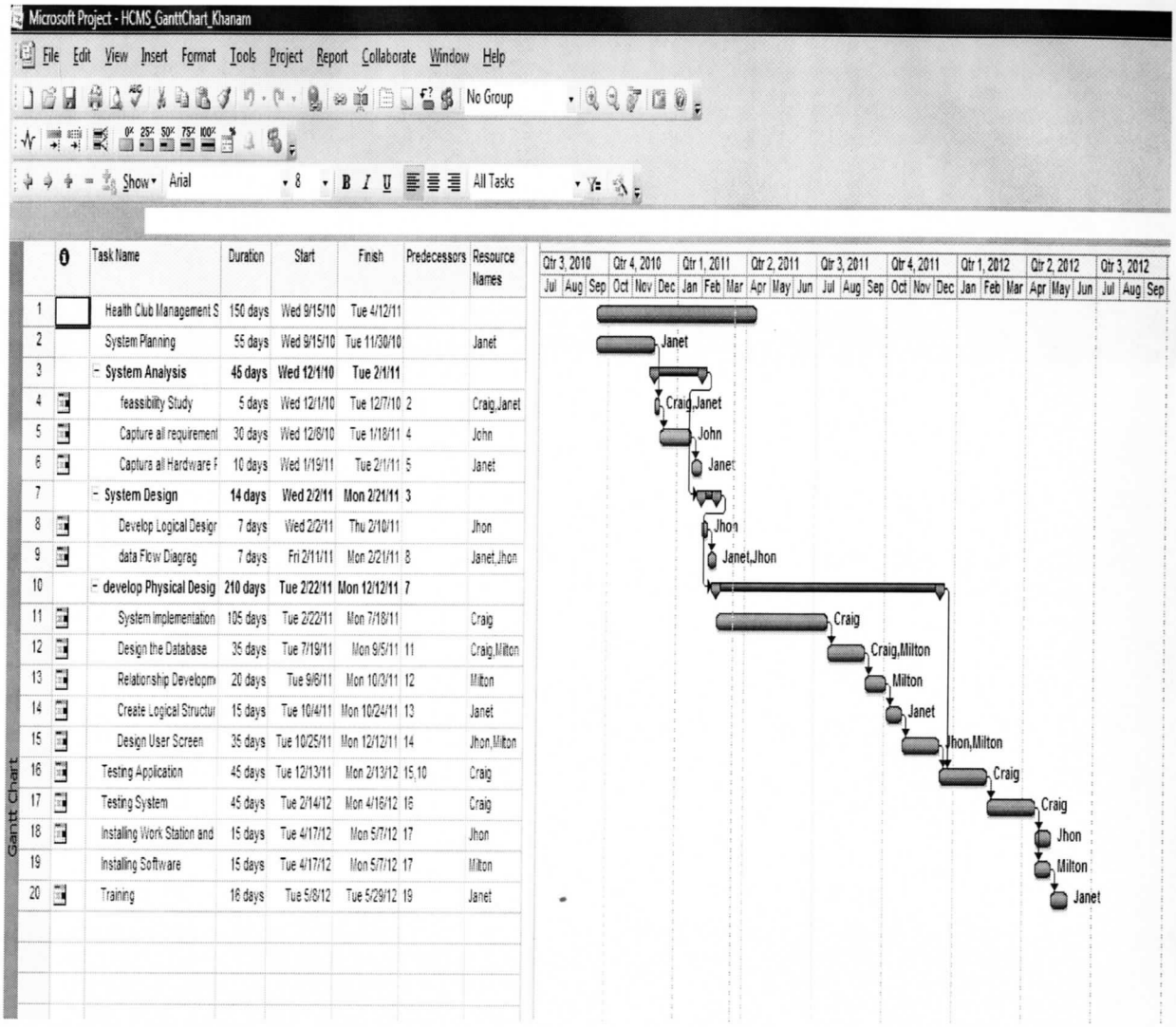


Figure 3-1 Gantt chart

CHAPTER 4

PROJECT DESIGN

4.1 N-Tier Architecture

The project is designed based on N-Tier architecture. The N-Tier Architecture divides the application processing two or more computers. The servers can be a database servers, application servers, or management servers. Scalability is one of the greatest advantages of N-Tier Architectures over traditional client server architecture. Scalability is the main issue in client –server Architecture but in N-tier architecture as the demand for application changes the backend systems can be easily scaled to the requirements. In this Environment the clients connect to servers over a network and using network protocols. This environment uses the graphical user interface where the interface is easier to understand and learn, and also offers more flexibility.

The N-tier architecture also uses single common user interface to communicate with the server known as Browser. Having so many advantages we have decided to implement the project in N-Tier Architecture. The following section describes the details of the various tiers in this project. The project architecture is in three different layers.

- Presentation Layer
- Business Layer
- Data Layer

4.1.1 Presentation Layer:

Presentation Layer provides the software interface that end-users see in static html and JavaScript page. The entire presentation layer is presented through a simple web browser. The web browser gives users access to the World Wide Web, provides graphical interface to navigate between the different pages of the application, and to send requests to the web servers. The hypertext markup language is a coded format language used for presenting the information in web browser. JavaScript is used in the presentation layer to perform client side validation. JavaScript is a popular scripting developed by Netscape; JavaScript programs can be included within an HTML document and can be used to make the static HTML pages more interactive

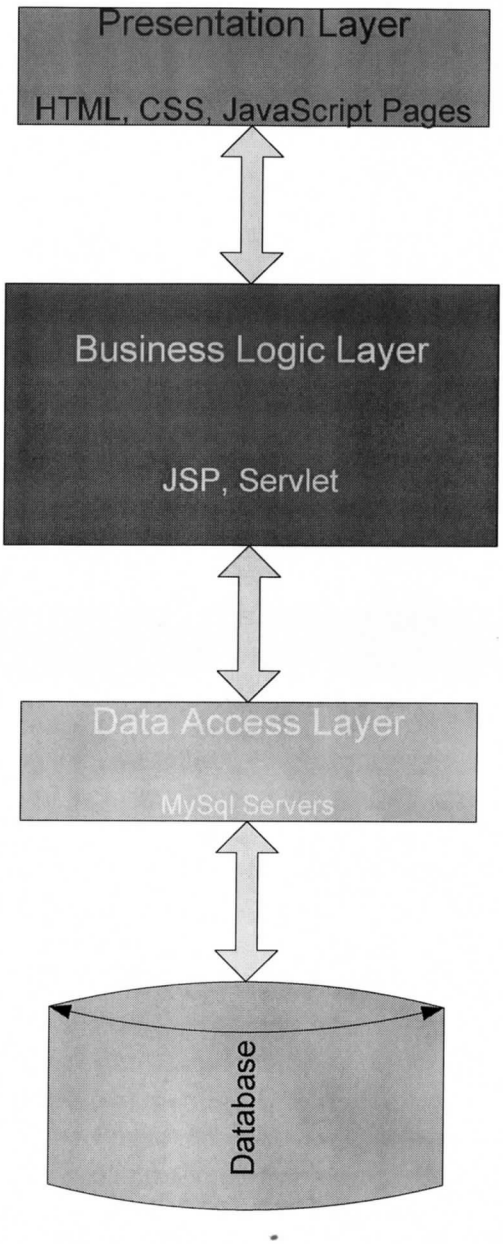


Figure 4-1 Technological components

4.1.2 Business Layer:

The business layer is pulled out from the presentation tier and, as its own layer, it controls the application's functionality by performing detailed processing. Java, JSP, Servlet has been used to develop business layer. In business layer all the business

logic is implemented. It is acting as the medium of communication between presentation layer and data layer.

SQL: Structured Query Language (SQL) is an interactive programming language used to send requests to the database. Using SQL data in the database can be easily manipulated and retrieved. SQL is also accepted universally by almost all Databases.

The programmed code of the presentation layer is presented in APPENDIX-A.

4.1.3 Data Layer:

The data layer provides the storage platform. It contains the physical data that is used throughout the system. The data layer consists of a built in Microsoft MySQL Server 5.5. The DBMS in the data layer receives calls from the Business Layer and processes those calls and send back the appropriate data to the presentation layer.

MySQL Server 5.5: MySQL Server is database management system, which accepts user calls in the form of SQL and processes those user calls and sends back the response to user. MySQL Server is a very efficient and modern database management system, which provides many state of the art features like Advanced Replication, and Distributed data processing.

4.2 Requirement Analysis:

The Requirement Analysis Process is the combination of tasks to understand the needs of the HCM and HCM members and expectations from the proposed system. This document will be focused on identifying stakeholders/actors, measurable goals and use cases to determine Member, performance and design requirements of HCM System.

4.2.1 Actors/Goals list

The primary stakeholders of the HCM System are the Members. All other stakeholders include people who have interest in the system, such as business owners and HCM staff members. Table 4.1 below describes all the actors who will interact with the system and their goals of interest.

Table 4.1 list of the actors and their goal for the proposed system

	Actor	Goals
1	HCM staff	Maintain Member Information
2	HCM staff	Maintain Trainers Information
3	HCM staff	Maintain Staffs Information
4	HCM staff	Maintain Training Information
5	HCM staff	Payment Process
6	HCM staff	Register Member into the System
7	HCM staff	Register Trainer into the System
8	Member	Member Log into the System
9	Trainer	Trainer Log into the System
10	HCM staff	Staff log in to the System
11	Member	Search Available Training Session
12	HCM staff	Schedule Training
13	HCM staff	Generate Reports on Members
14	HCM Staff	Renew Membership

4.3 Use Case Description

Use Case descriptions are simple descriptions of the proposed system's functionality. The use cases are the backbone of the functional development. All UML diagrams created in this project are based on these use cases. The development team has used feedback from thousands of questionnaires completed by potential and existing Members to develop these use cases.

USE CASE 1: Maintain Member Information

Brief description
This use case allows HCM staff to add, delete and edit member information.

Primary Actor
HCM staff

Stakeholders
HCM staff

Member

Trigger
HCM staff wishes to maintain member information.

Preconditions
HCM staff logged in into the system.

Guarantees
Success End Condition

HCM staff successfully can add, delete and edit member information.

Failed End Condition

HCM staff fails to add, delete and edit member information.

MAIN SUCCESS SCENARIO

1. The system presents HCM staff with member information maintenance options

2a. HCM staff request to 'Add member'

3. HCM staff submits member information.

4. The system validates member information according to member information rules.

5a. The system saves member information, generates Member ID and presents HCM staff with a confirmation.

EXTENSIONS

*a. HCM staff decides to quit:

*a1. System asks HCM staff to save changes (if any)

*a1a. HCM staff chooses to save:

System saves changes and exits [success]

*a1b. HCM staff chooses to quit:

System discards any intermediate data and exits [fail]

2b. HCM staff request to 'Retrieve member information':

2b1. HCM staff submits search request.

2b2. System searches for member.

2b2a. Member not found

System notifies HCM staff [repeat]

2b2b. Member found:

System presents member information.

2c. HCM staff request to 'Update member information':

2c1. HCM staff submits search request.

2c2. System searches for member.

2c2a. Member not found

System notifies HCM staff [repeat]

2c2b. Member found:

System presents member information.

HCM staff provides updated information.

The system validates member information according to member information rules.

Invalid member information:

System notifies HCM staff [repeat]

The system saves member information and presents HCM staff with a confirmation.

2d. HCM staff request to 'Delete member information':

2d1. HCM staff submits search request.

2d2. System searches for member.

2d2a. Member not found

System notifies HCM staff [repeat]

2d2b. Member found:

System presents member information.

HCM staff confirms it is the desired member and selects to delete member.

The system confirms that HCM staff wants to delete member.

Cancel deletes operation:

System aborts the delete operation and notifies HCM staff [repeat]

The system deletes the member and presents HCM staff with a confirmation.

5b. Invalid member information:

5b1. System notifies HCM staff [repeat]

USE CASE 2: Maintain Trainers Information

Brief description
This use case allows HCM staff to add, delete and edit trainer information.

Primary Actor
HCM staff

Stakeholders
HCM staff

Trainer

Trigger
HCM staff wishes to maintain trainer information.

Preconditions
HCM staff logged in into the system.

Guarantees
Success End Condition

HCM staff successfully can add, delete and edit trainer information.

Failed End Condition

HCM staff fails to add, delete and edit trainer information.

MAIN SUCCESS SCENARIO

1. The system presents HCM staff with trainer information maintenance options

2a. HCM staff request to 'Add trainer'

3. HCM staff submits trainer information.

4. The system validates trainer information according to trainer information rules.

5a. The system saves trainer information, generates trainer ID and presents HCM staff with a confirmation.

EXTENSIONS
*a. HCM staff decides to quit:

*a1. System asks HCM staff to save changes (if any)

*a1a. HCM staff chooses to save:

System saves changes and exits [success]

*a1b. HCM staff chooses to quit:

System discards any intermediate data and exits [fail]

2b. HCM staff request to 'Retrieve trainer information':

2b1. HCM staff submits search request.

2b2. System searches for trainer.

2b2a. Trainer not found

System notifies HCM staff [repeat]

2b2b. Trainer found:

System presents trainer information.

2c. HCM staff request to 'Update trainer information':

2c1. HCM staff submits search request.

2c2. System searches for trainer.

2c2a. Trainer not found

System notifies HCM staff [repeat]

2c2b. Trainer found:

System presents trainer information.

HCM staff provides updated information.

The system validates trainer information according to trainer information rules.

Invalid trainer information:

System notifies HCM staff [repeat]

The system saves trainer information and presents HCM staff with a confirmation.

2d. HCM staff request to 'Delete trainer information':

2d1. HCM staff submits search request.

2d2. System searches for trainer.

2d2a. Trainer not found

System notifies HCM staff [repeat]

2d2b. Trainer found:

System presents trainer information.

HCM staff confirms it is the desired trainer and selects to delete trainer.

The system confirms that HCM staff wants to delete trainer.

Cancel deletes operation:

System aborts the delete operation and notifies HCM staff [repeat]

The system deletes the trainer and presents HCM staff with a confirmation.

5b. Invalid trainer information:

5b1. System notifies HCM staff [repeat]

USE CASE 3: Maintain Staffs Information

Brief description
This use case allows HCM staff to add, delete and edit staff information.

Primary Actor
HCM staff

Stakeholders
HCM staff

Trigger
HCM staff wishes to maintain staff information.

Preconditions
HCM staff logged in into the system.

Guarantees
Success End Condition

HCM staff successfully can add, delete and edit staff information.

Failed End Condition

HCM staff fails to add, delete and edit staff information.

MAIN SUCCESS SCENARIO

1. The system presents HCM staff with staff information maintenance options

2a. HCM staff request to 'Add staff'

3. HCM staff submits staff information.

4. The system validates staff information according to staff information rules.

5a. The system saves staff information, generates staff ID and presents HCM staff with a confirmation.

EXTENSIONS
*a. HCM staff decides to quit:

*a1. System asks HCM staff to save changes (if any)

*a1a. HCM staff chooses to save:

System saves changes and exits [success]

*a1b. HCM staff chooses to quit:

System discards any intermediate data and exits [fail]

2b. HCM staff request to 'Retrieve staff information':

2b1. HCM staff submits search request.

2b2. System searches for staff.

2b2a. Staff not found

System notifies HCM staff [repeat]

2b2b. Staff found:

System presents staff information.

2c. HCM staff request to 'Update staff information':

2c1. HCM staff submits search request.

2c2. System searches for staff.

2c2a. Staff not found

System notifies HCM staff [repeat]

2c2b. Staff found:

System presents staff information.

HCM staff provides updated information.

The system validates staff information according to staff information rules.

Invalid staff information:

System notifies HCM staff [repeat]

The system saves staff information and presents HCM staff with a confirmation.

2d. HCM staff request to 'Delete staff information':

2d1. HCM staff submits search request.

2d2. System searches staff.

2d2a. Staff not found

System notifies HCM staff [repeat]

2d2b. Staff found:

System presents staff information.

HCM staff confirms it is the desired staff and selects to delete staff.

The system confirms that HCM staff wants to delete staff.

Cancel deletes operation:

System aborts the delete operation and notifies HCM staff [repeat]

The system deletes the staff and presents HCM staff with a confirmation.

5b. Invalid staff information:

5b1. System notifies HCM staff [repeat]

USE CASE 4: Maintain Training Information

Brief description
This use case allows HCM staff to add, delete and edit training information.

Primary Actor
HCM staff

Stakeholders
HCM staff

Trainer

Member

Trigger
HCM staff wishes to maintain training information.

Preconditions
HCM staff logged in into the system.

Guarantees
Success End Condition

HCM staff successfully can add, delete and edit training information.

Failed End Condition

HCM staff fails to add, delete and edit training information.

MAIN SUCCESS SCENARIO

1. The system presents HCM staff with training information maintenance options

2a. HCM staff request to 'Add training'

3. HCM staff submits training information.

4. The system validates training information according to training information rules.

5a. The system saves training information, generates training ID and presents HCM staff with a confirmation.

EXTENSIONS
*a. HCM staff decides to quit:

*a1. System asks HCM staff to save changes (if any)

*a1a. HCM staff chooses to save:

System saves changes and exits [success]

*a1b. HCM staff chooses to quit:

System discards any intermediate data and exits [fail]

2b. HCM staff request to 'Retrieve training information':

2b1. HCM staff submits search request.

2b2. System searches for training.

2b2a. Training not found

System notifies HCM staff [repeat]

2b2b. Training found:

System presents training information.

2c. HCM staff request to 'Update training information':

2c1. HCM staff submits search request.

2c2. System searches for training.

2c2a. Training not found

System notifies HCM staff [repeat]

2c2b. Training found:

System presents training information.

HCM staff provides updated information.

The system validates training information according to training information rules.

Invalid training information:

System notifies HCM staff [repeat]

The system saves training information and presents HCM staff with a confirmation.

2d. HCM staff request to 'Delete training information':

2d1. HCM staff submits search request.

2d2. System searches for training.

2d2a. Training not found

System notifies HCM staff [repeat]

2d2b. Training found:

System presents training information.

HCM staff confirms it is the desired training and selects to delete training.

The system confirms that HCM staff wants to delete training.

Cancel deletes operation:

System aborts the delete operation and notifies HCM staff [repeat]

The system deletes the training and presents HCM staff with a confirmation.

5b. Invalid training information:

5b1. System notifies HCM staff [repeat]

USE CASE 5: Payment Process

Brief description
This use case allows HCM staff to process the payments by Member.

Primary Actor
HCM staff

Stakeholders
HCM staff
Member

Trigger
The HCM staff wishes to process payment.

Preconditions
HCM staff logged in into the system.

Guarantees
Success End Condition

The system properly process the payment

Failed End Condition

Nothing occurs

MAIN SUCCESS SCENARIO
1. The HCM staff enters the Invoice Number and payment details into the system

2a. The system validate the payment method

3. HCM staff selects "Process Payment"

4a. The system generate a Transaction Number and records the transaction information

5.The system generates and prints a receipt

EXTENSIONS
2b. The system fails to validate the payment

2.b.1. The HCM staff enters alternate payment information

2.b.2 The system validates the secondary payment type

4b. The system fails to record the transaction information.

USE CASE 6: Register Member into the System

Brief description

This use case allows HCM staff to register member into the system.

Primary Actor

HCM staff

Stakeholders

HCM staff

Member

Trigger

HCM staff wishes to register new member into the system.

Preconditions

HCM staff logged in into the system.

Guarantees

Success End Condition

HCM staff registers new member into the site.

Failed End Condition

System fails to register Member.

MAIN SUCCESS SCENARIO

1. HCM staff selects 'Add new Member' option.
2. System presents registration form to HCM staff.
3. HCM staff enters member information into member registration form.

4a. System validates input information.

5a. System saves Member information.

6a. System sends a confirmation to HCM staff.

EXTENSIONS

4b. System fails to validate information.

4b1. HCM staff enters correct information.

Repeats step 3-4 until all field are valid.

5b. System fails to save Member information.

5b1. System notifies HCM staff and exit [fail]

6b. System fails to sends a confirmation to HCM staff.

USE CASE 7: Register Trainer into the System

Brief description

This use case allows HCM staff to register trainer into the system.

Primary Actor

HCM staff

Stakeholders

HCM staff

Trainer

Trigger

HCM staff wishes to register into the system.

Preconditions

HCM staff logged in into the system

Guarantees

Success End Condition

HCM staff registers trainer into the site.

Failed End Condition

System fails to register trainer.

MAIN SUCCESS SCENARIO

1. HCM staff selects 'Add New Trainer' option.
2. System presents registration form to HCM staff.
3. HCM staff enters trainer information into trainer registration form.

4a. System validates input information.

5a. System saves trainer information.

6a. System sends a confirmation to HCM staff.

EXTENSIONS

4b. System fails to validate information.

4b1. HCM staff enters correct information.

Repeats step 3-4 until all field are valid.

5b. System fails to save trainer information.

5b1. System notifies HCM staff and exit [fail]

6b. System fails to sends a confirmation to HCM staff.

USE CASE 8: Member Log into the System

Brief description

This use case allows a Member to log into the system.

Primary Actor

Member

Stakeholders

HCM staff

Member

Trigger

Member wishes to log in into the system.

Preconditions

Member is already registered with HCM system.

Guarantees

Success End Condition

Member logged in into the system.

Failed End Condition

Member fails to log in into the system

MAIN SUCCESS SCENARIO

1. Member selects to log in into the system.
2. System presents log in form to Member.
3. Member enters user name and password into the system.

4a. System validates Member username and password.

5. System directs Member to member page.

EXTENSIONS

4b. System fails to validate Member information.

4b1. System show failure message.

USE CASE 9: Trainer Log into the System

Brief description

This use case allows a trainer to log into the system.

Primary Actor

Trainer

Stakeholders

CSS staff

Trainer

Trigger

Trainer wishes to log in into the system.

Preconditions

Trainer is already registered with HCM system.

Guarantees

Success End Condition

Trainer logged in into the system.

Failed End Condition

Trainer fails to log in into the system

MAIN SUCCESS SCENARIO

1. Trainer selects to log in into the system.
2. System presents log in form to Trainer.
3. Trainer enters user name and password into the system.

4a. System validates trainer username and password.

5. System directs trainer to trainer page.

EXTENSIONS

4b. System fails to validate trainer information.

4b1. System show failure message.

USE CASE 10: Staff log into the System

Brief description

This use case allows a staff to log into the system.

Primary Actor

HCM Staff

Stakeholders

HCM Staff

Trigger

Staff wishes to log in into the system.

Preconditions

Staff is already registered with HCM system.

Guarantees

Success End Condition

Staff logged in into the system.

Failed End Condition

Staff fails to log in into the system

MAIN SUCCESS SCENARIO

1. Staff selects to log in into the system.
2. System presents log in form to staff.
3. Staff enters user name and password into the system.

4a. System validates staff username and password.

5. System directs staff to staff page.

EXTENSIONS

4b. System fails to validate staff information.

4b1. System show failure message.

USE CASE 11: Search Available Training Session

Brief description

This use case allows the Member to search for training session.

Primary Actor

Member

Stakeholders

HCM staff

Member

Trigger

Member wishes to search available training session.

Preconditions

The Member is on the HCM website homepage.

Guarantees

Success End Condition

Member searches available training session.

Failed End Condition

Member fails to search available training session.

MAIN SUCCESS SCENARIO

1. Member selects to search available training session option.
2. System presents search form with available criteria.

3a. Member choose search by training type.

4a. System shows a list of available training with matched search criteria.

EXTENSIONS

3b. Member choose search by date.

3c. Member choose search by duration.

4b. System shows available service not available.

USE CASE 12: Schedule Training

Brief description

This use case allows the HCM staff to schedule a personal trainer.

Primary Actor

HCM staff

Stakeholders

HCM staff

Member

Trainer

Trigger

HCM staff wishes to schedule training.

Preconditions

HCM staff logged in into the system

Guarantees

Success End Condition

HCM staff schedules training.

Failed End Condition

HCM staff fails to schedule training.

MAIN SUCCESS SCENARIO

1. HCM staff selects "Schedule Training" option.
2. System presents Schedule Training form.
3. HCM staff enters trainer information into Schedule Training form.

4a. System validates input information.

5a. System saves Schedule Training information.

6a. System sends a confirmation to HCM staff.

EXTENSIONS

4b. System fails to validate information.

4b1. HCM staff enters correct information.

Repeats step 3-4 until all field are valid.

5b. System fails to save Schedule Training information.

5b1. System notifies HCM staff and exit [fail]

6b. System fails to sends a confirmation to HCM staff.

USE CASE 13: Generate Reports on Members

Brief description

This use case allows the staff to generate reports on current member.

Primary Actor

Member

Stakeholders

HCM staff

Trigger

HCM staff wishes to generate reports on current member.

Preconditions

The staff is logged in to the system.

Guarantees

Success End Condition

HCM staff generates reports on current member.

Failed End Condition

HCM staff fails to generate reports on current member.

MAIN SUCCESS SCENARIO

1. Staff selects to search members report option.

2. System presents search form with available criteria.

3a. Staff choose search by report type.

4a. System generates report on available search criteria.

EXTENSIONS

3b. Staff choose search by member ship type.

3c. Staff choose search by member payment information.

3d. Staff choose search by member ship ending date.

4b. System shows available service not available.

Use case 14: Renew Membership

Brief description
This use case allows the HCM staff to renew membership.

Primary Actor
HCM staff

Stakeholders
HCM staff
Member

Trigger
HCM staff wishes to renew membership.

Preconditions
The HCM staff logged into the system.

Guarantees
Success End Condition

The HCM staff can renew membership.

Failed End Condition

System fails to renew membership.

MAIN SUCCESS SCENARIO

1. HCM staff enters member ID information to the system

2. System presents member information to HCM staff.

3a. HCM staff selects update account information with "Renewed".

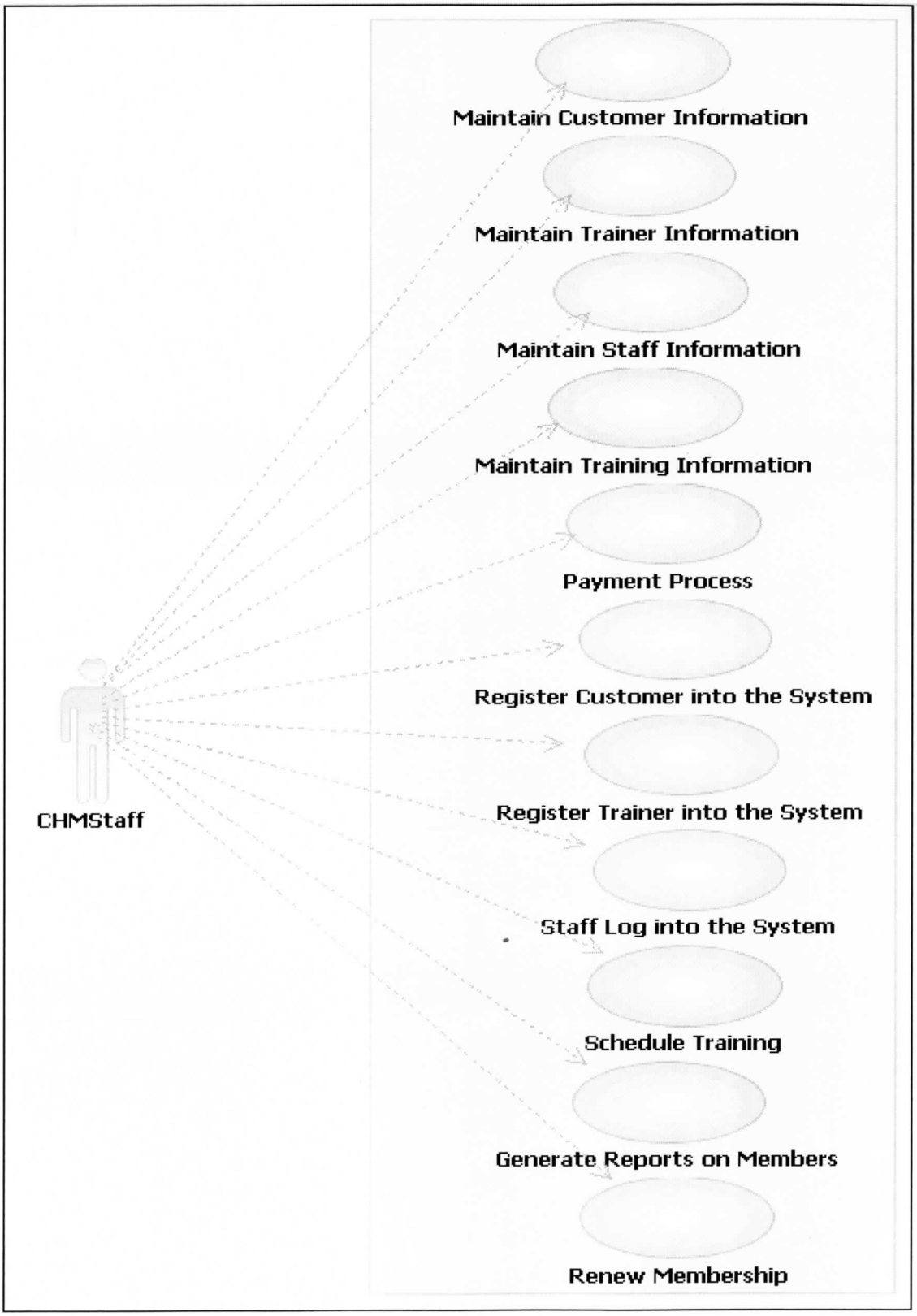
4a. System calculate "early renewal discount" and updates account information.

Extensions

3b. HCM staff quit.

4b. System fails to update account information.

4.4 Use Case Diagram



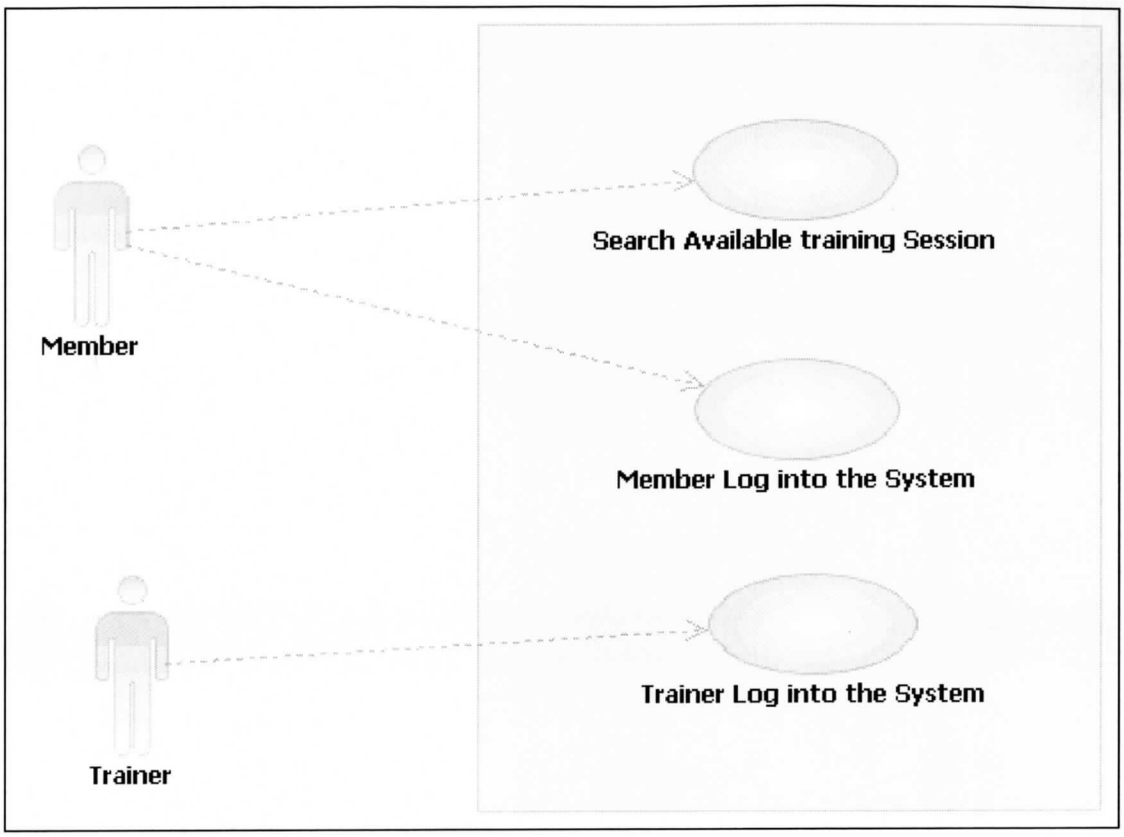


Figure 4-2 Use Case Diagram

4.5 Domain Model

A domain model illustrates important conceptual classes in a problem domain. In this case, Figure 4.5 illustrates the conceptual classes (a real world concept or thing i.e. vacation package) of the newly proposed system (problem domain). The domain model is important to the development team because it helps them to understand the structure of data that supports the business process. The domain model also helps the development team to create a vocabulary to use during the development process that ultimately leads to closing the "semantic gap" between the real world and the software world.

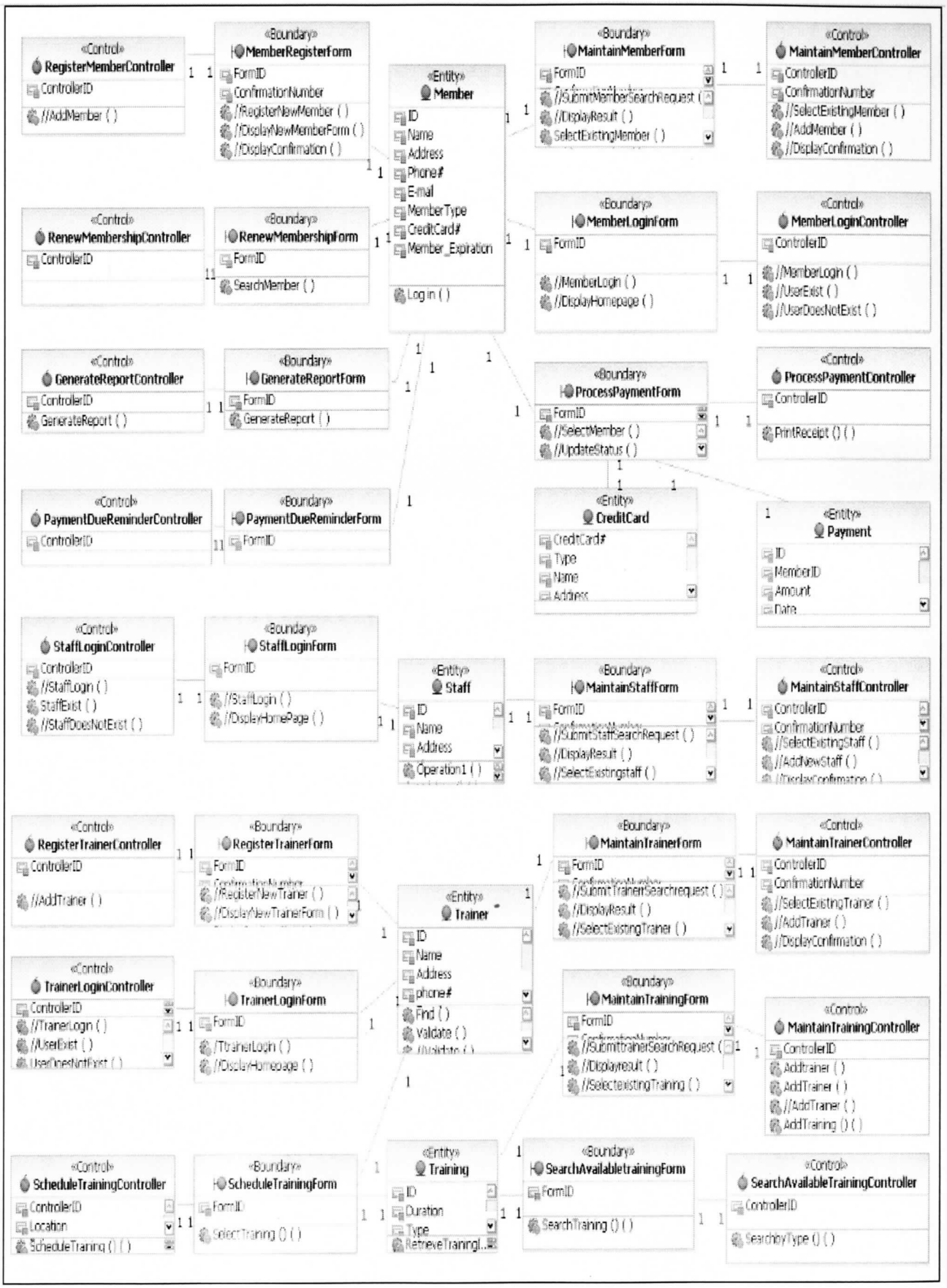

«Control»
RegisterMemberController
ControlerID
//AddMember ( )
«Boundary»
MemberRegisterForm
FormID
ConfirmationNumber
/RegisterNewMember ( )
/DisplayNewMemberForm ( )
/DisplayConfirmation ( )
«Entity»
Member
ID
Name
Address
Phone#
E-mail
MemberType
CreditCard#
Member_Expiration
Log in ( )
«Boundary»
MaintainMemberForm
FormID
//SubmitMemberSearchRequest ( )
//DisplayResult ( )
SelectExistingMember ( )
«Control»
MaintainMemberController
ControlerID
ConfirmationNumber
//SelectExistingMember ( )
//AddMember ( )
//DisplayConfirmation ( )
«Control»
RenewMembershipController
ControlerID
«Boundary»
RenewMembershipForm
FormID
SearchMember ( )
«Boundary»
MemberLoginForm
FormID
//MemberLogin ( )
//DisplayHomepage ( )
«Control»
MemberLoginController
ControlerID
//MemberLogin ( )
//UserExist ( )
//UserDoesNotExist ( )
«Control»
GenerateReportController
ControlerID
GenerateReport ( )
«Boundary»
GenerateReportForm
FormID
GenerateReport ( )
«Boundary»
ProcessPaymentForm
FormID
//SelectMember ( )
//UpdateStatus ( )
«Control»
ProcessPaymentController
ControlerID
PrintReceipt ( ) ( )
«Control»
PaymentDueReminderController
ControlerID
«Boundary»
PaymentDueReminderForm
FormID
«Entity»
CreditCard
CreditCard#
Type
Name
Address
«Entity»
Payment
ID
MemberID
Amount
Date
«Control»
StaffLoginController
ControlerID
//StaffLogin ( )
StaffExist ( )
//StaffDoesNotExist ( )
«Boundary»
StaffLoginForm
FormID
//StaffLogin ( )
//DisplayHomePage ( )
«Entity»
Staff
ID
Name
Address
Operation1 ( )
«Boundary»
MaintainStaffForm
FormID
//SubmitStaffSearchRequest ( )
//DisplayResult ( )
//SelectExistingstaff ( )
«Control»
MaintainStaffController
ControlerID
ConfirmationNumber
//SelectExistingStaff ( )
//AddNewStaff ( )
«Control»
RegisterTrainerController
ControlerID
//AddTrainer ( )
«Boundary»
RegisterTrainerForm
FormID
/RegisterNewTrainer ( )
/DisplayNewTrainerForm ( )
«Entity»
Trainer
ID
Name
Address
phone#
Find ( )
Validate ( )
«Boundary»
MaintainTrainerForm
FormID
//SubmitTrainerSearchrequest ( )
//DisplayResult ( )
//SelectExistingTrainer ( )
«Control»
MaintainTrainerController
ControlerID
ConfirmationNumber
//SelectExistingTrainer ( )
//AddTrainer ( )
//DisplayConfirmation ( )
«Control»
TrainerLoginController
ControlerID
//TrainerLogin ( )
//UserExist ( )
«Boundary»
TrainerLoginForm
FormID
/TrainerLogin ( )
/DisplayHomepage ( )
«Boundary»
MaintainTrainingForm
FormID
//SubmittrainerSearchRequest ( )
//Displayresult ( )
//SelectexistingTraining ( )
«Control»
MaintainTrainingController
ControlerID
Addtrainer ( )
AddTrainer ( )
//AddTrainer ( )
AddTraining ( ) ( )
«Control»
ScheduleTrainingController
ControlerID
Location
ScheduleTraining ( ) ( )
«Boundary»
ScheduleTrainingForm
FormID
SelectTraining ( ) ( )
«Entity»
Training
ID
Duration
Type
«Boundary»
SearchAvailabletrainingForm
FormID
SearchTraining ( ) ( )
«Control»
SearchAvailableTrainingController
ControlerID
SearchbyType ( ) ( )

Figure 4-3 Domain Model

4.6 Use Case Realizations

The following diagrams were created to provide details and support the analysis and design of the project. The diagrams are organized into two sections, behavioral and structural.

4.6.1. Behavioral Models

4.6.1. Sequence Diagrams (CD) and Communication Diagrams (CD):

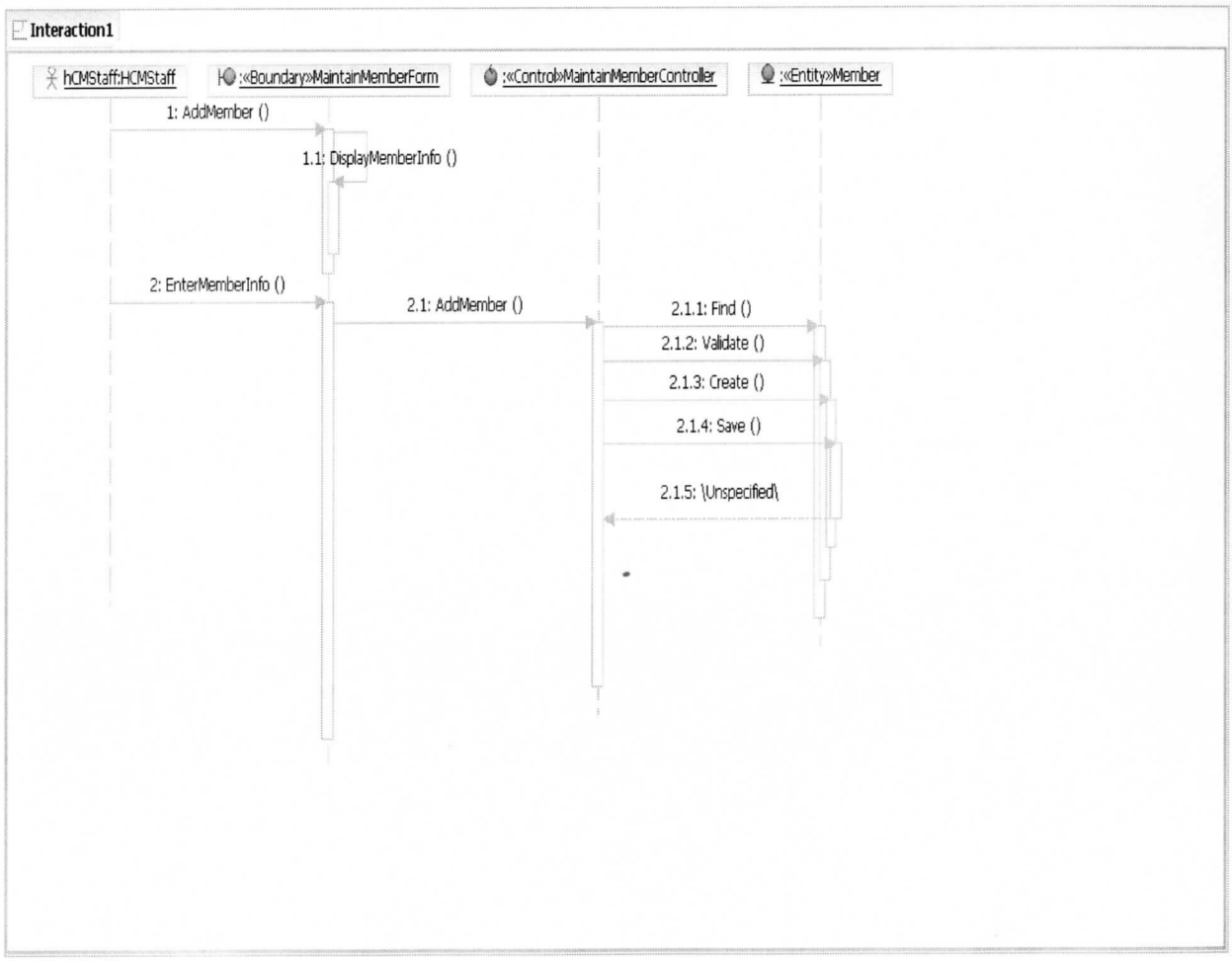


Figure 4-4(a) SD for Maintain Member

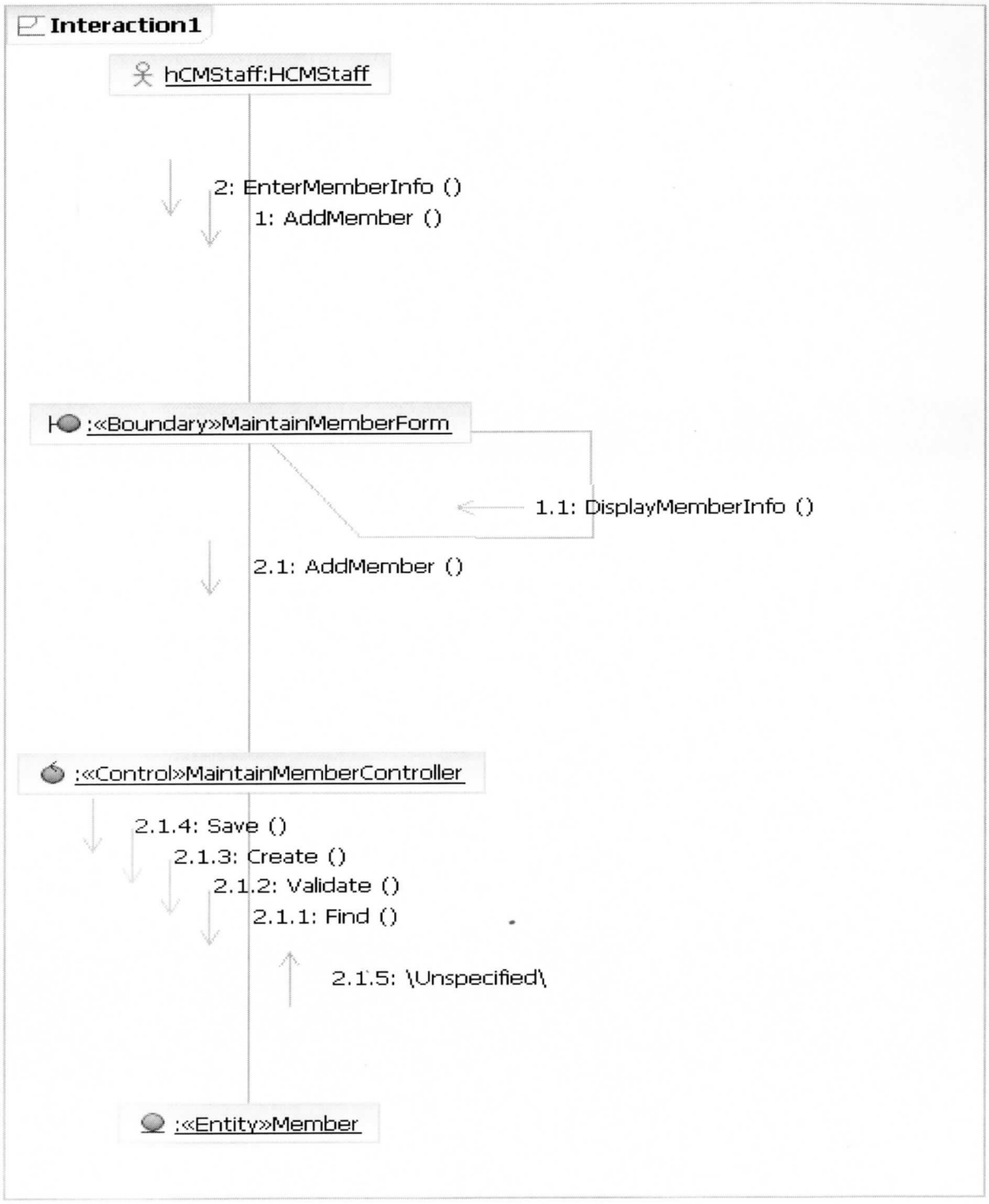


Figure 4-4 CD for Maintain Member

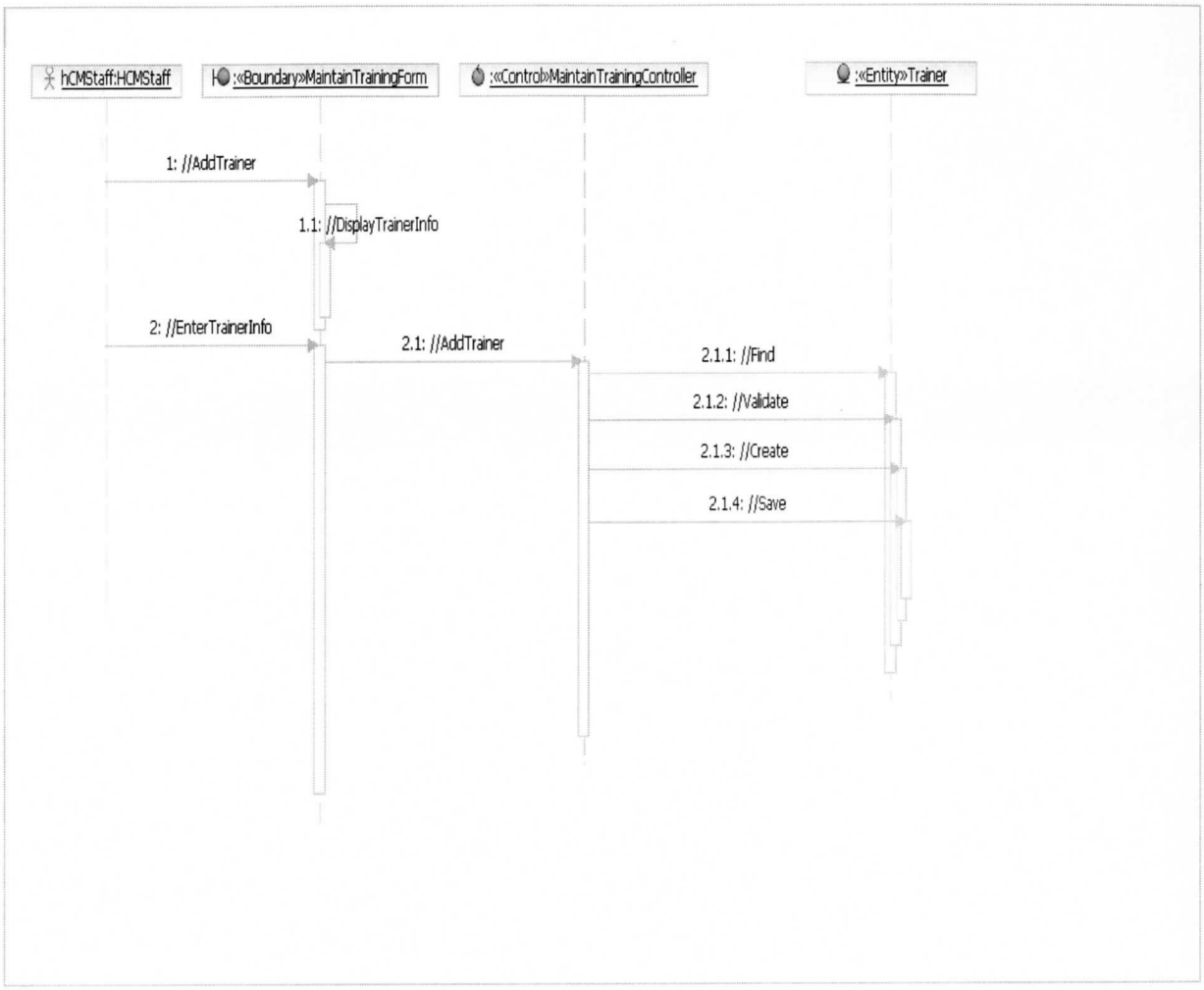


Figure 4-5 SD for Maintain Trainer

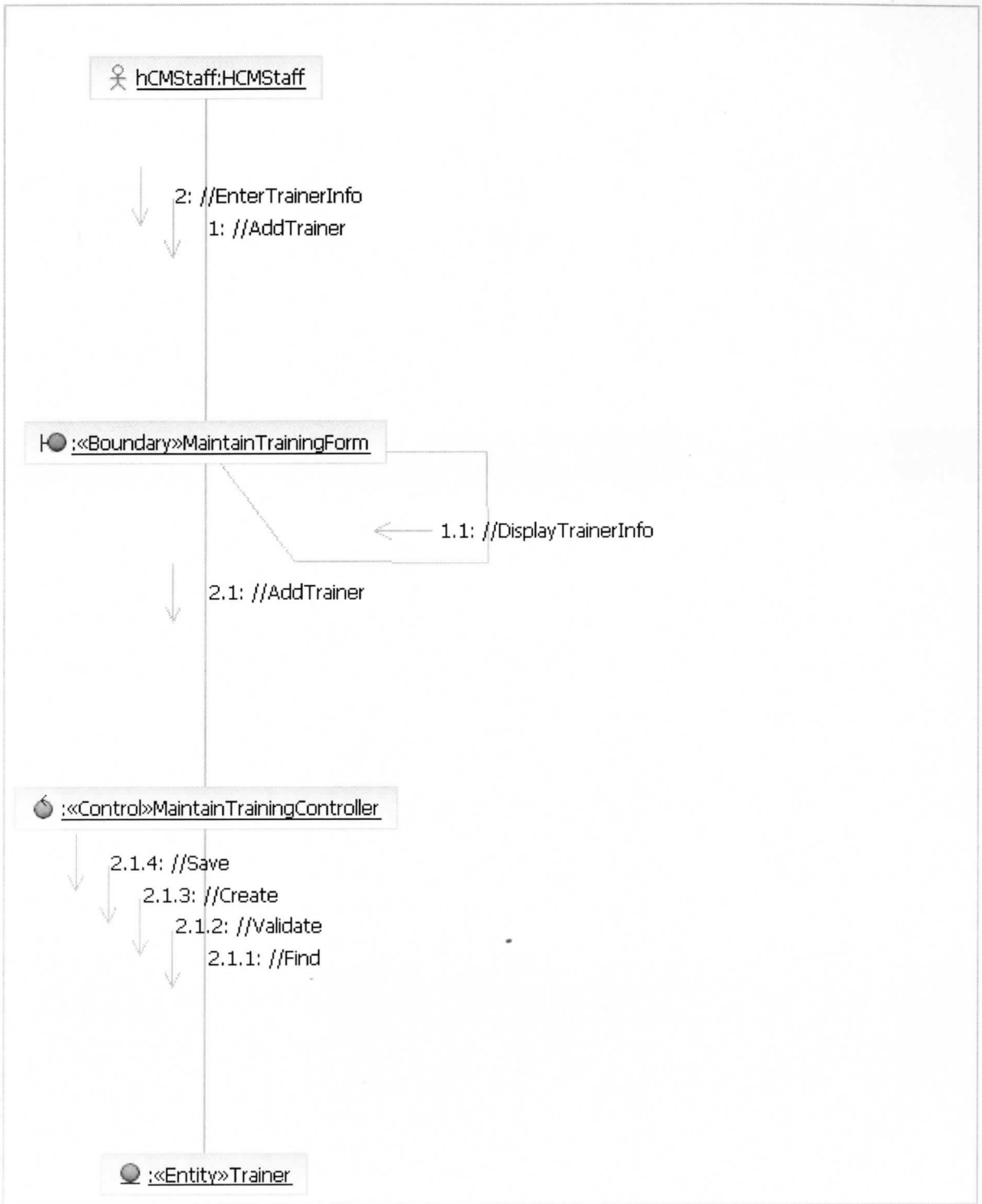


Figure 4-6 CD for Maintain Trainer

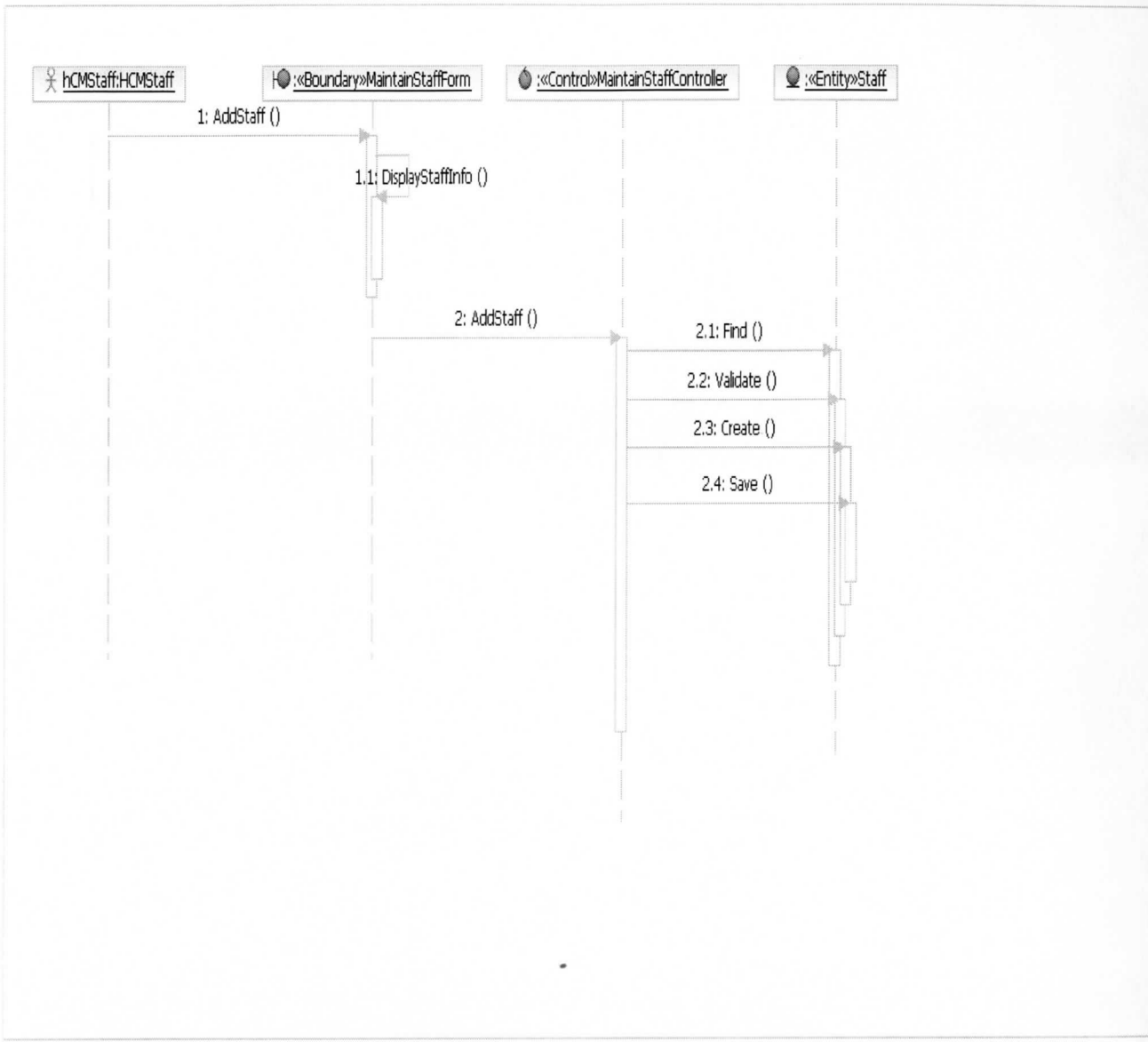


Figure 4-7 SD for Maintain Staff

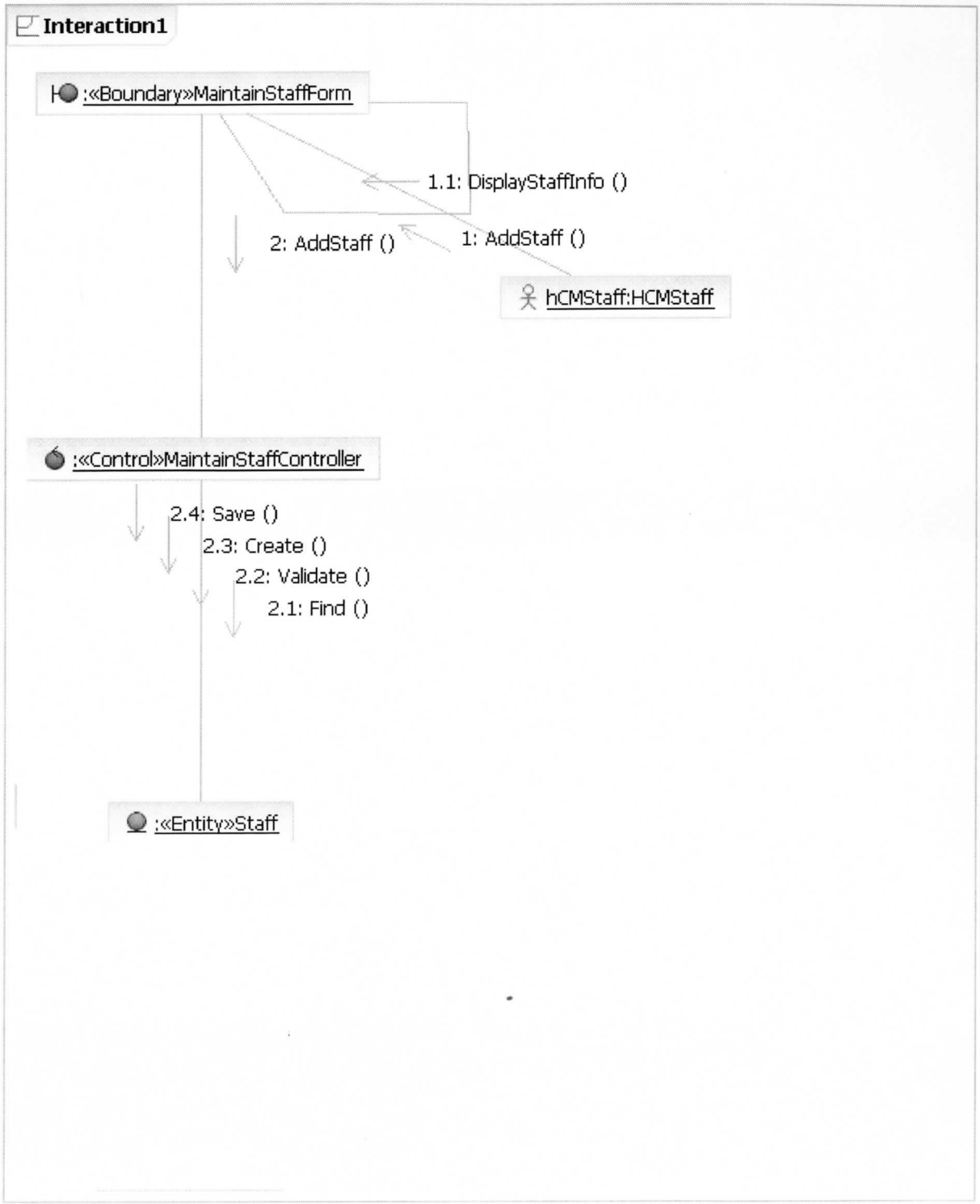


Figure 4-8 CD for Maintain Staff

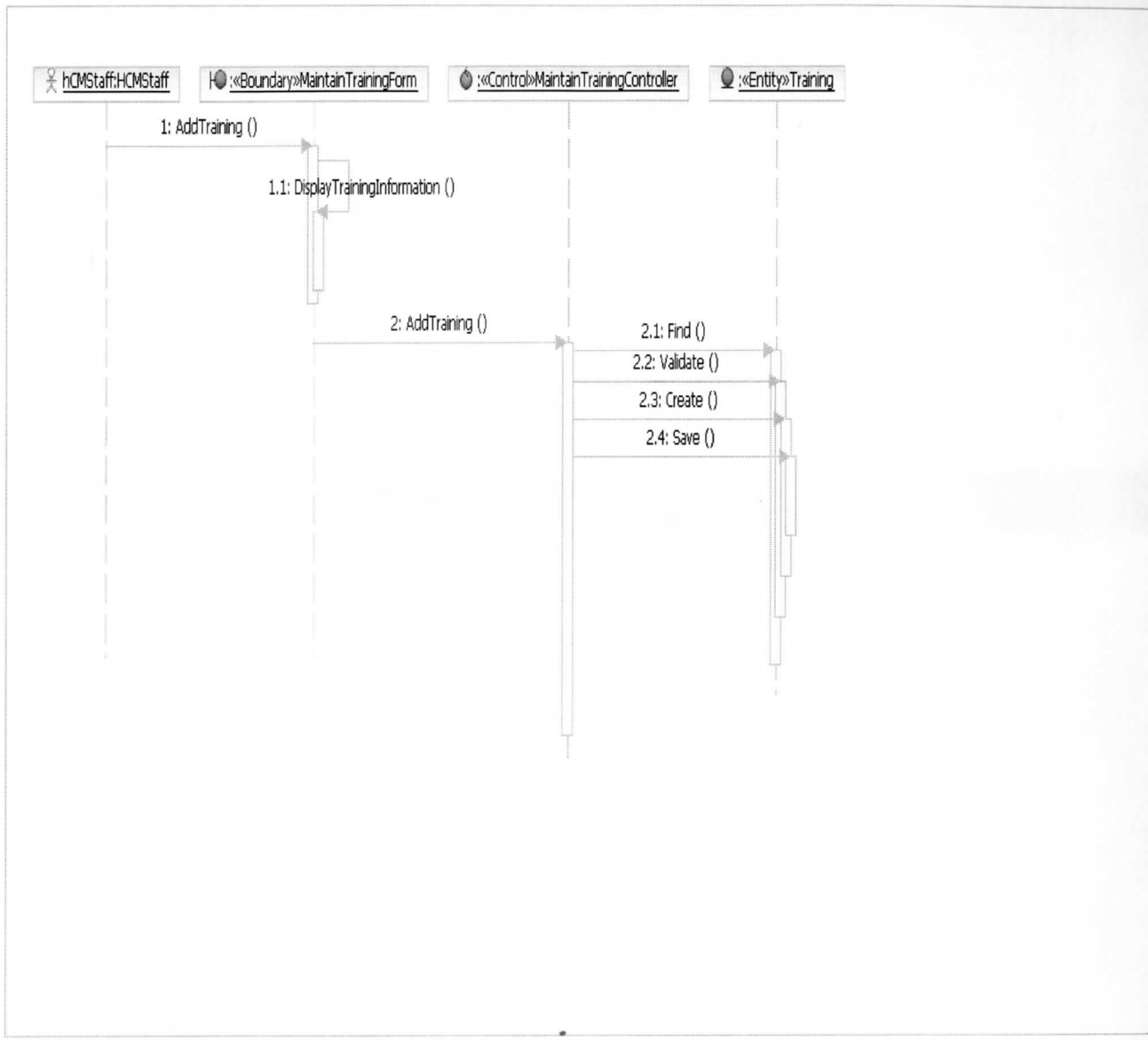


Figure 4-9 SD the Maintain Training

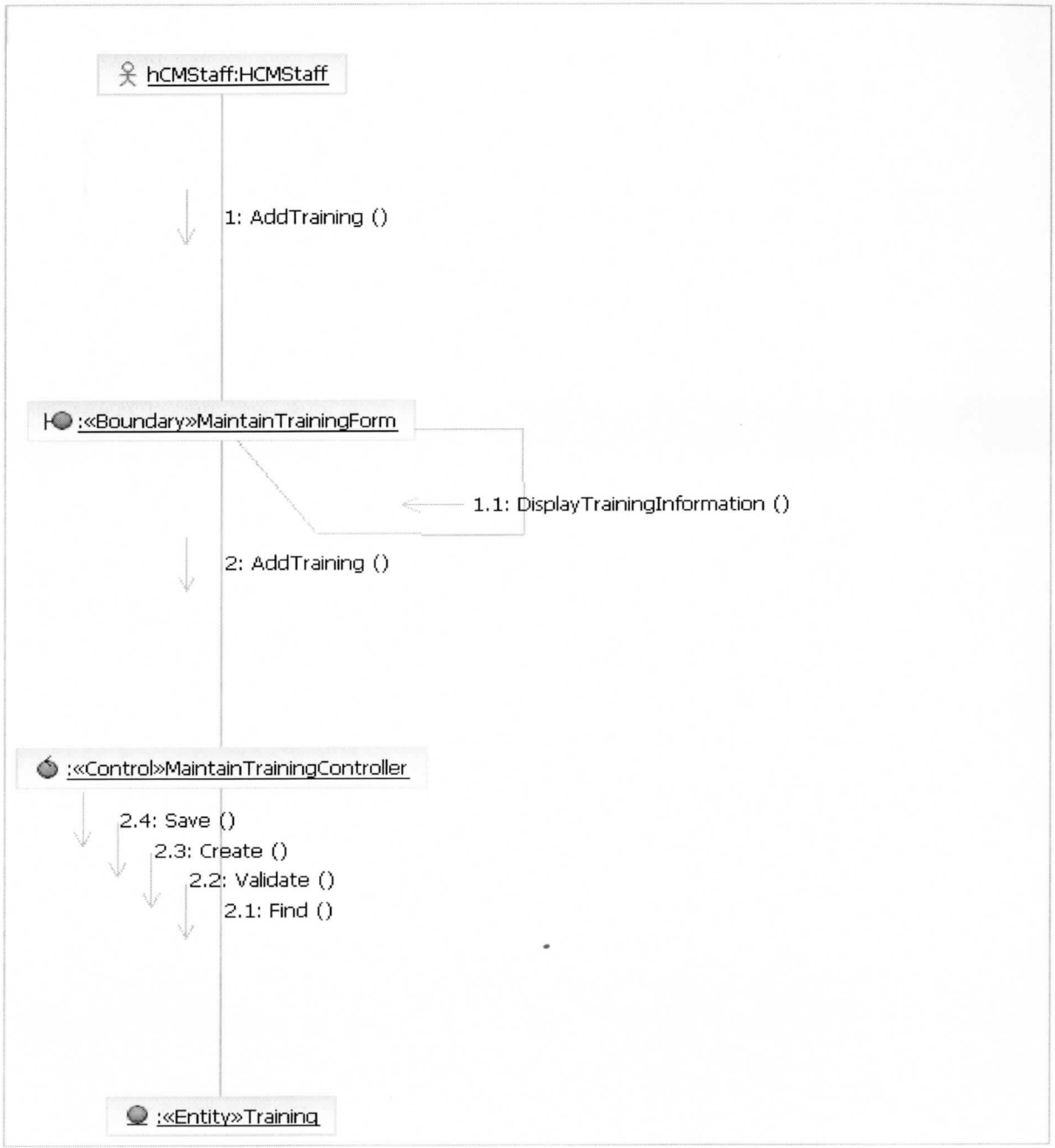


Figure 4-10 CD the Maintain Training

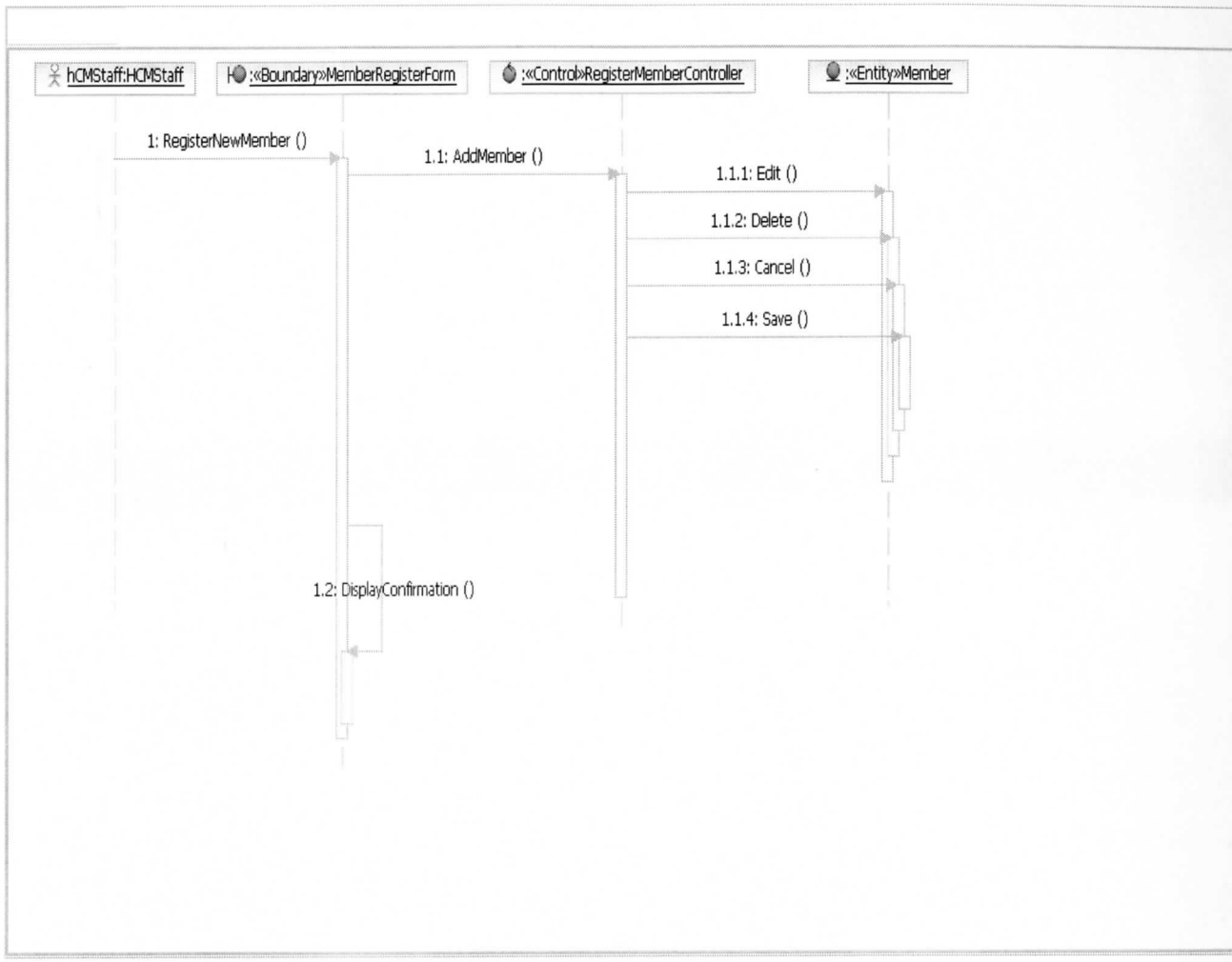


Figure 4-11 SD the Register Member

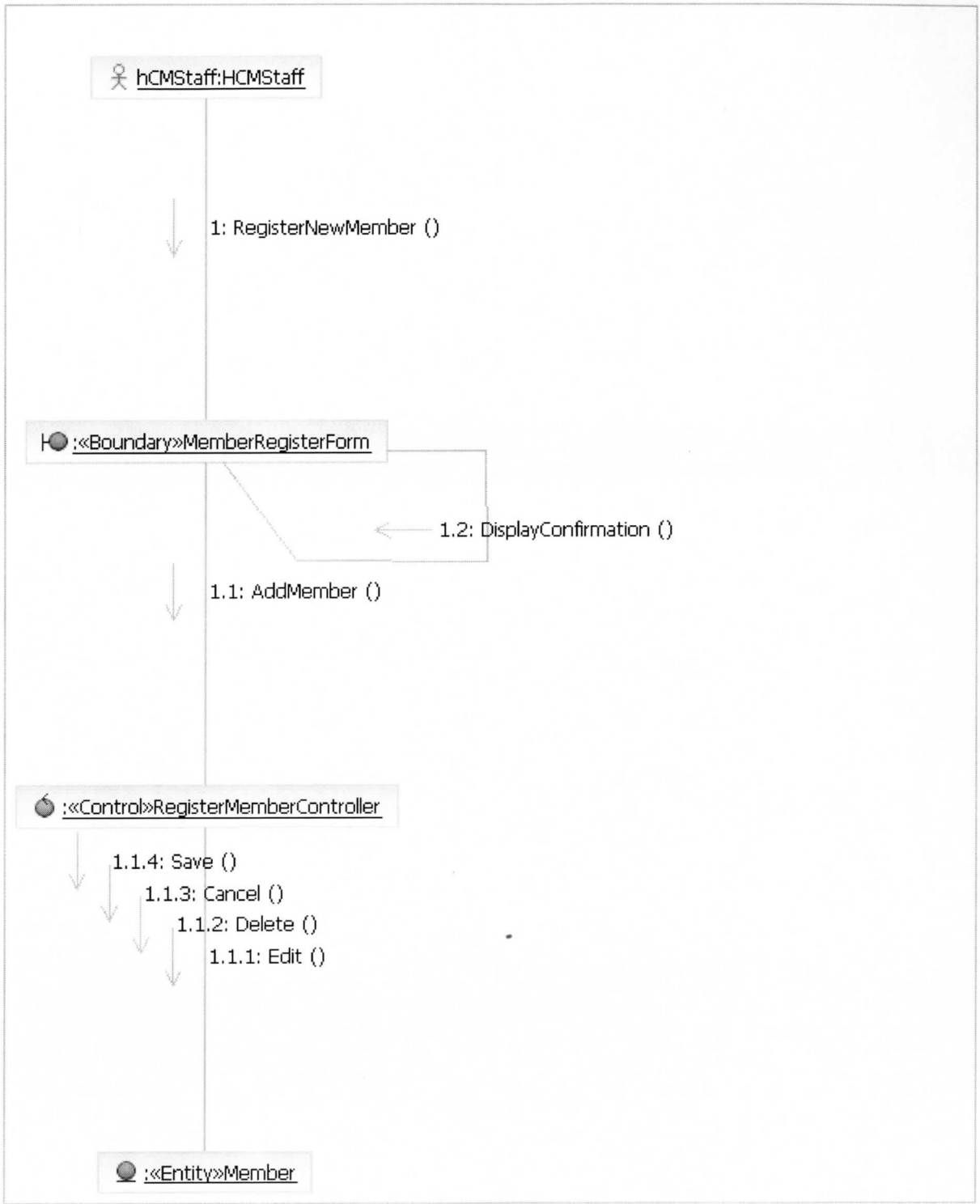


Figure 4-12 CD for Register Member

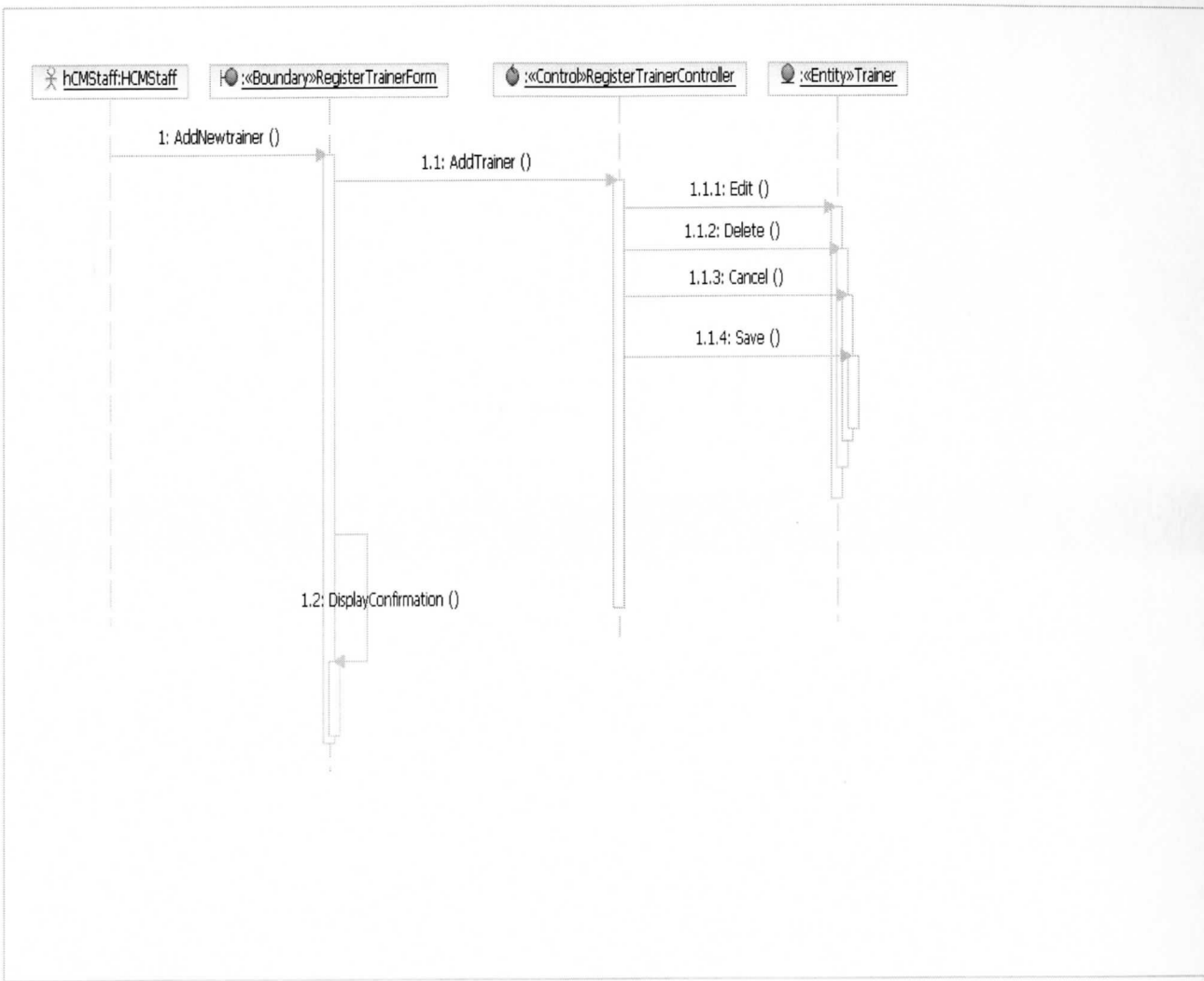


Figure 4-13 SD for Register Trainer

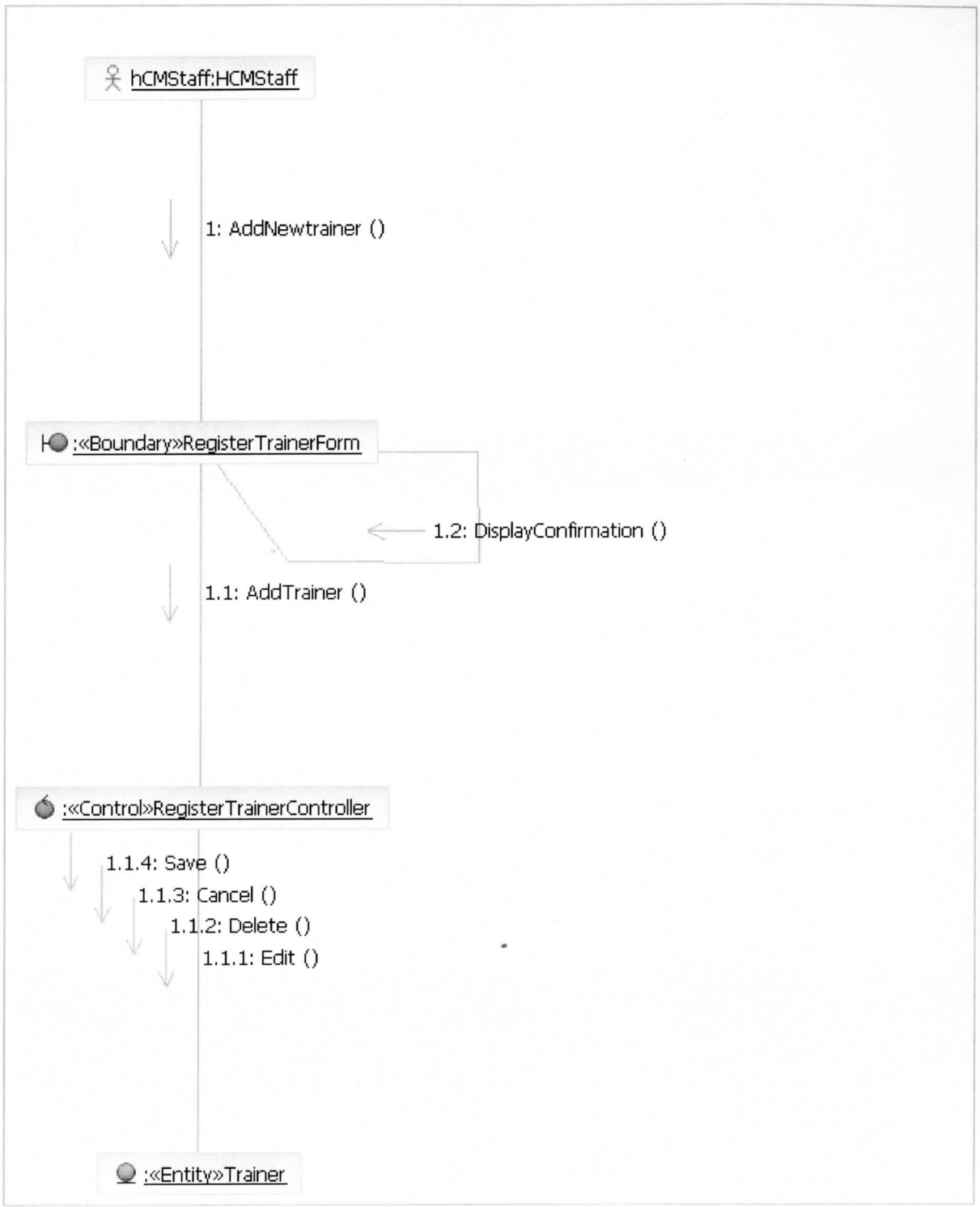


Figure 4-14 CD for Register Trainer

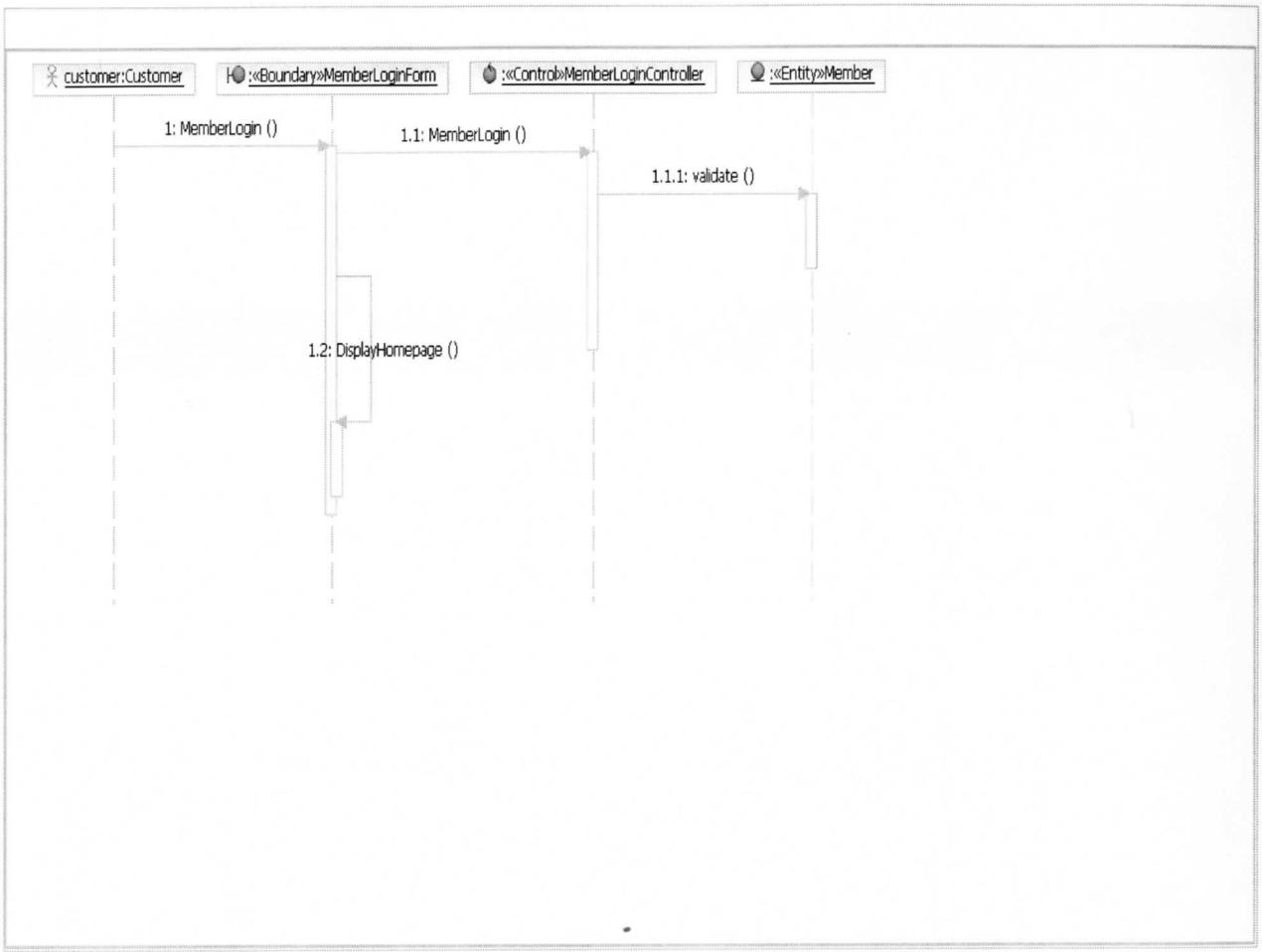


Figure 4-15 SD for Member Login

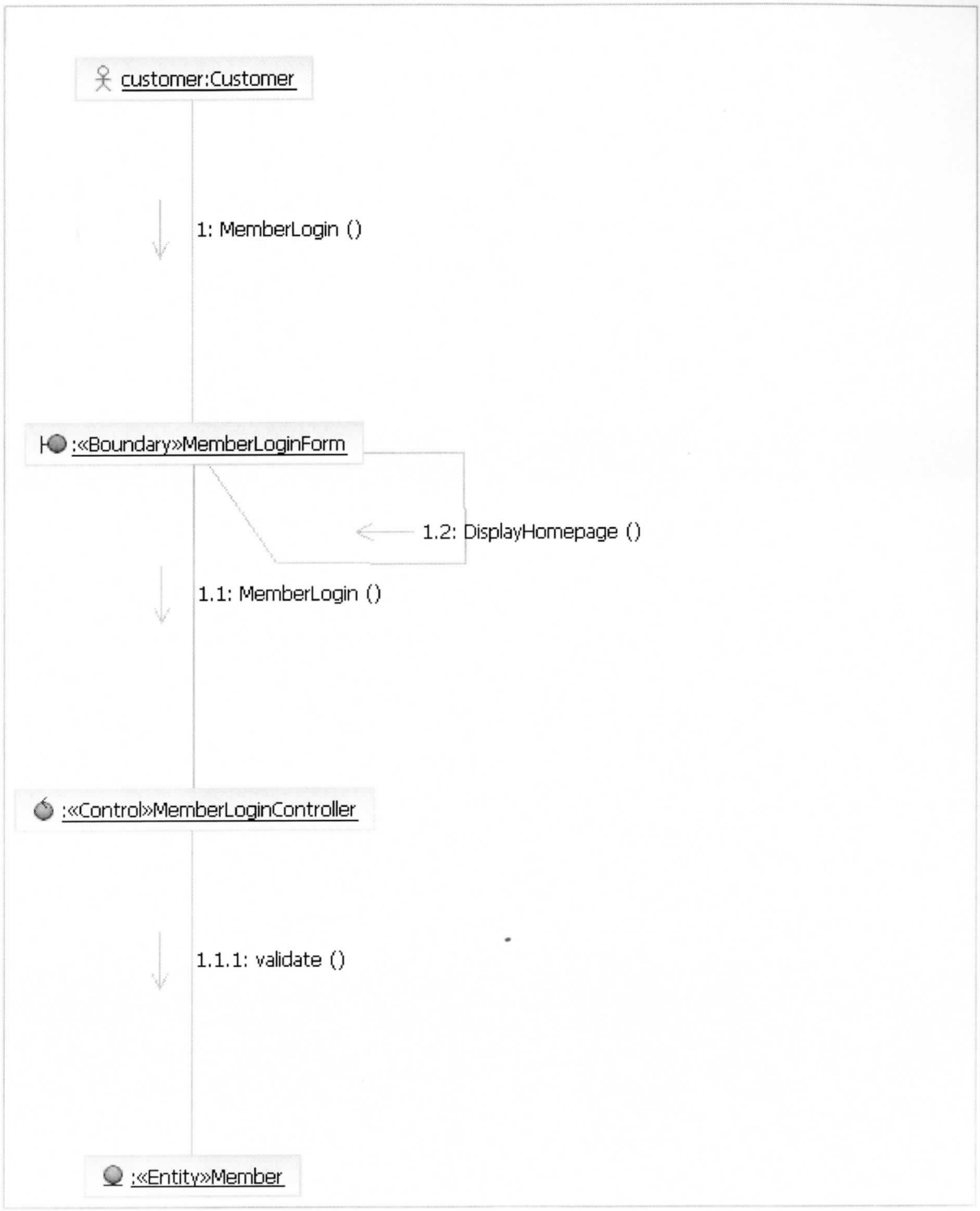


Figure 4-16 CD for Member Login

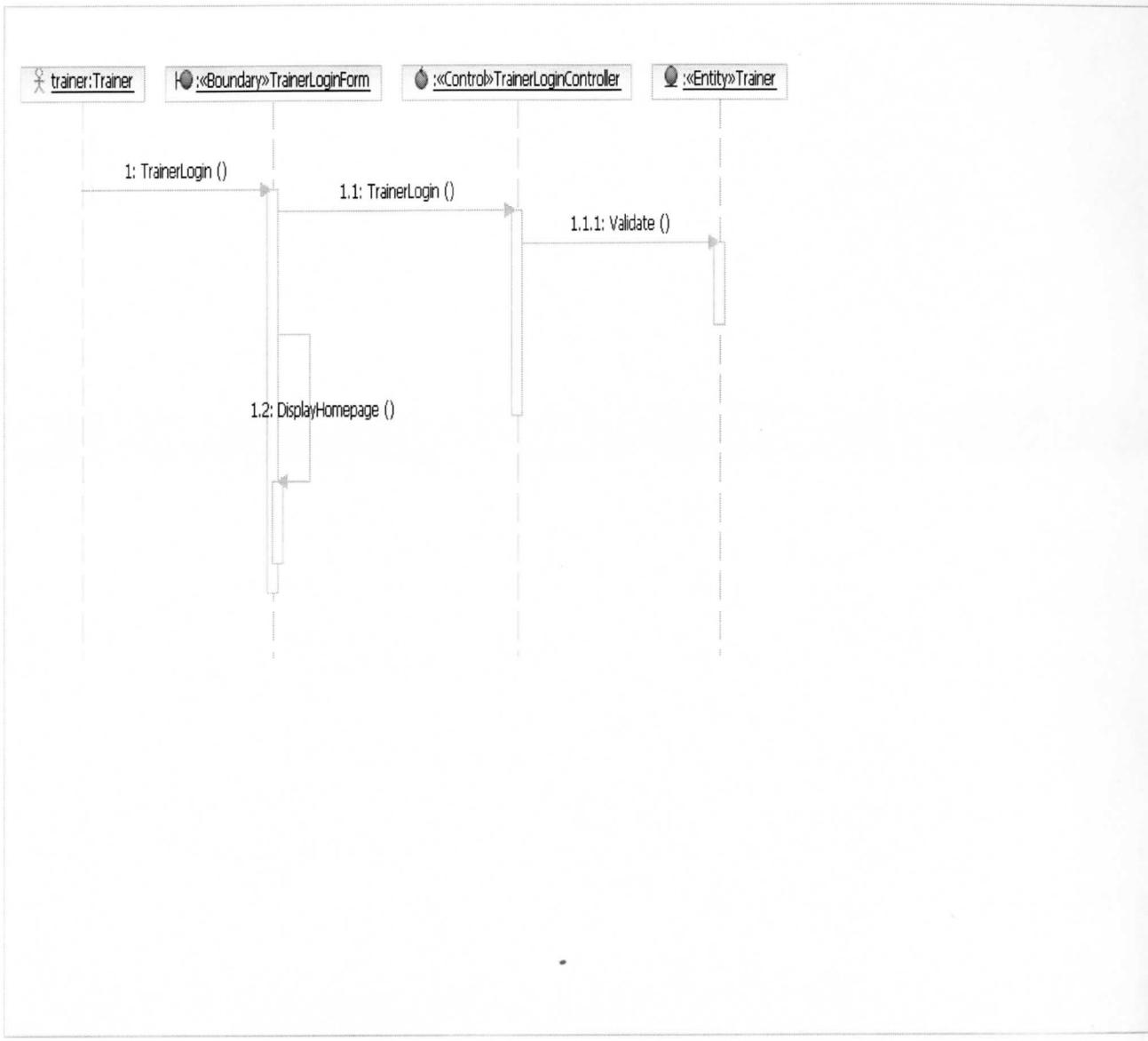


Figure 4-17 SD for Trainer Login

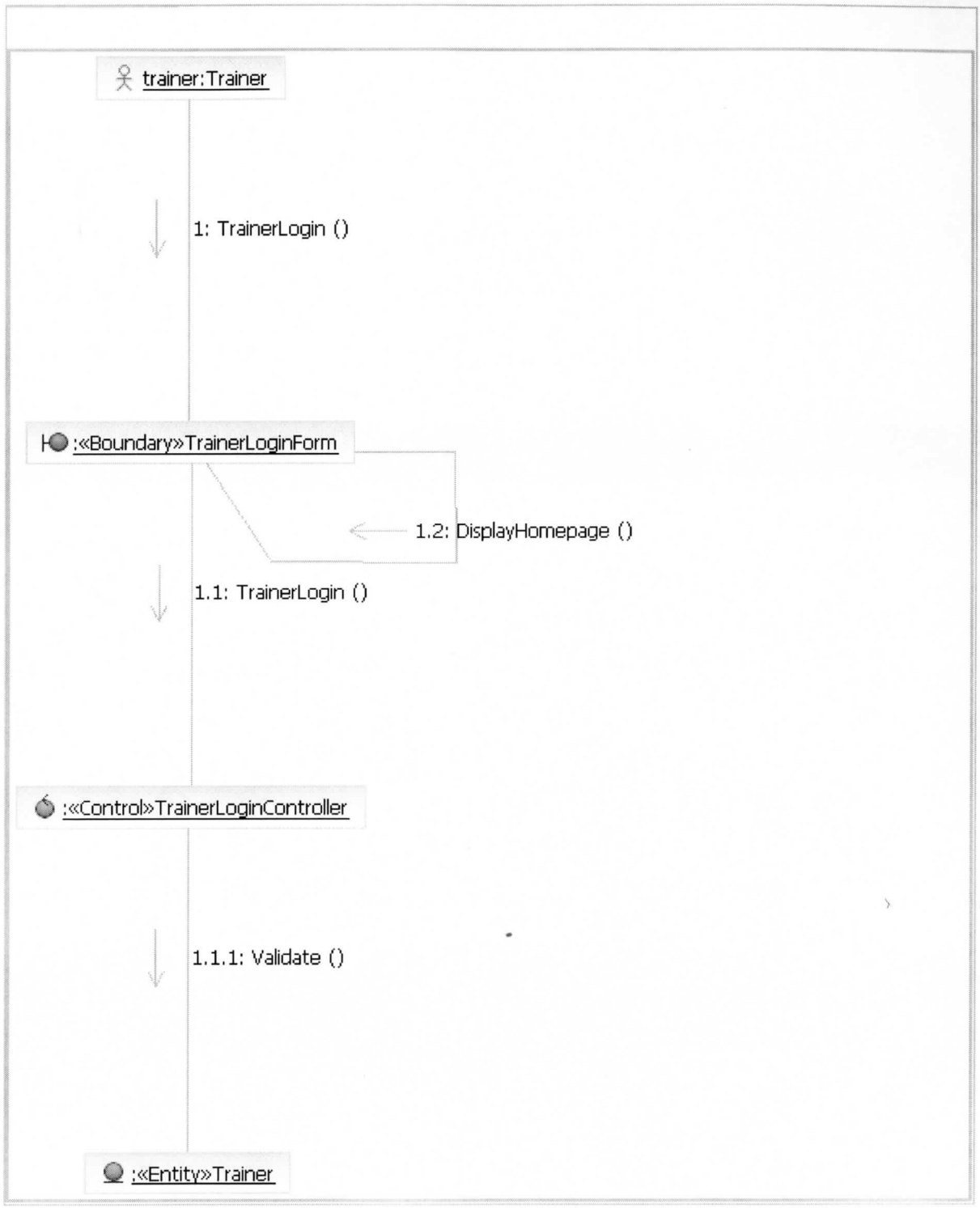


Figure 4-18 CD for Trainer Login

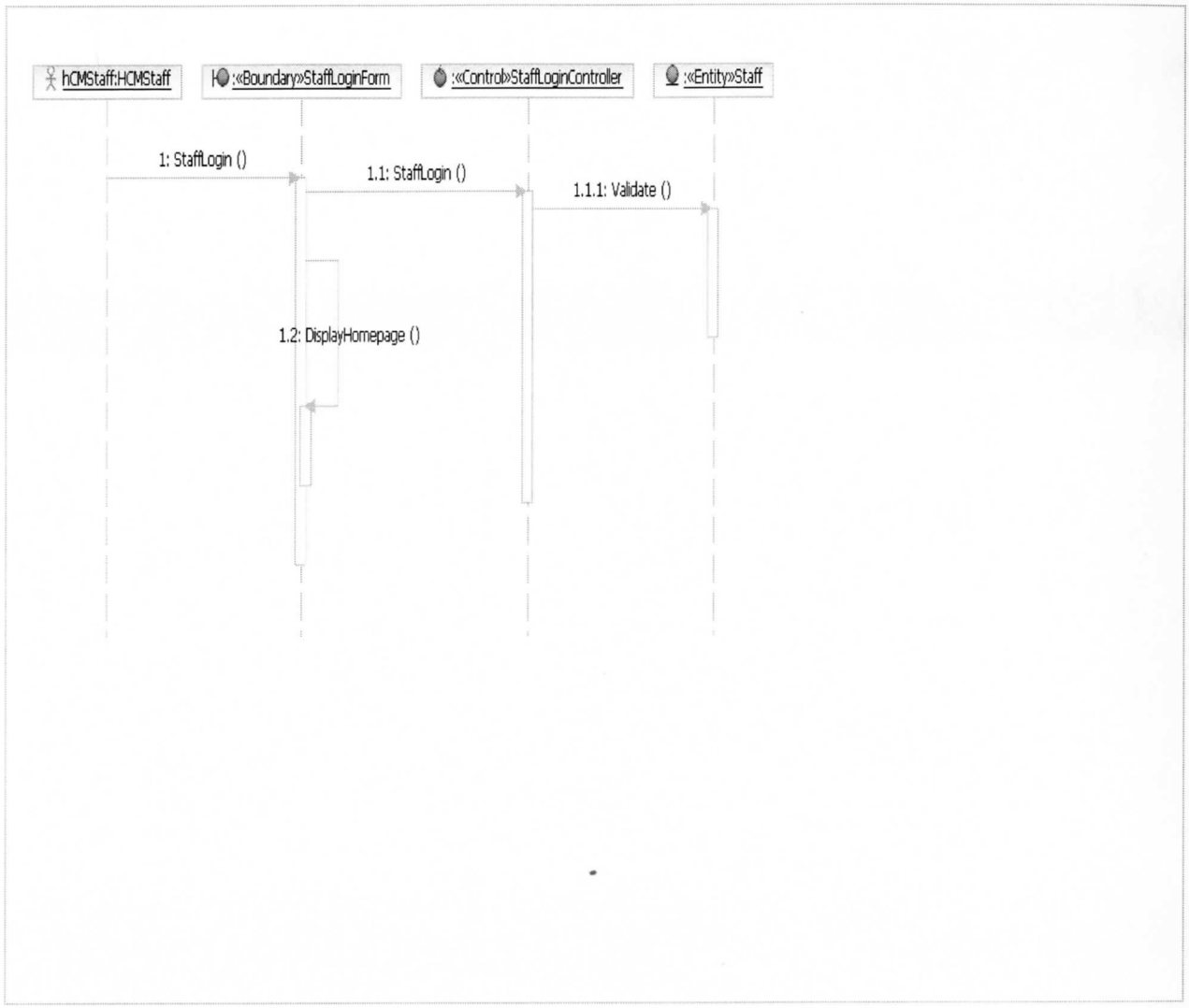


Figure 4-19 SD for Staff Login

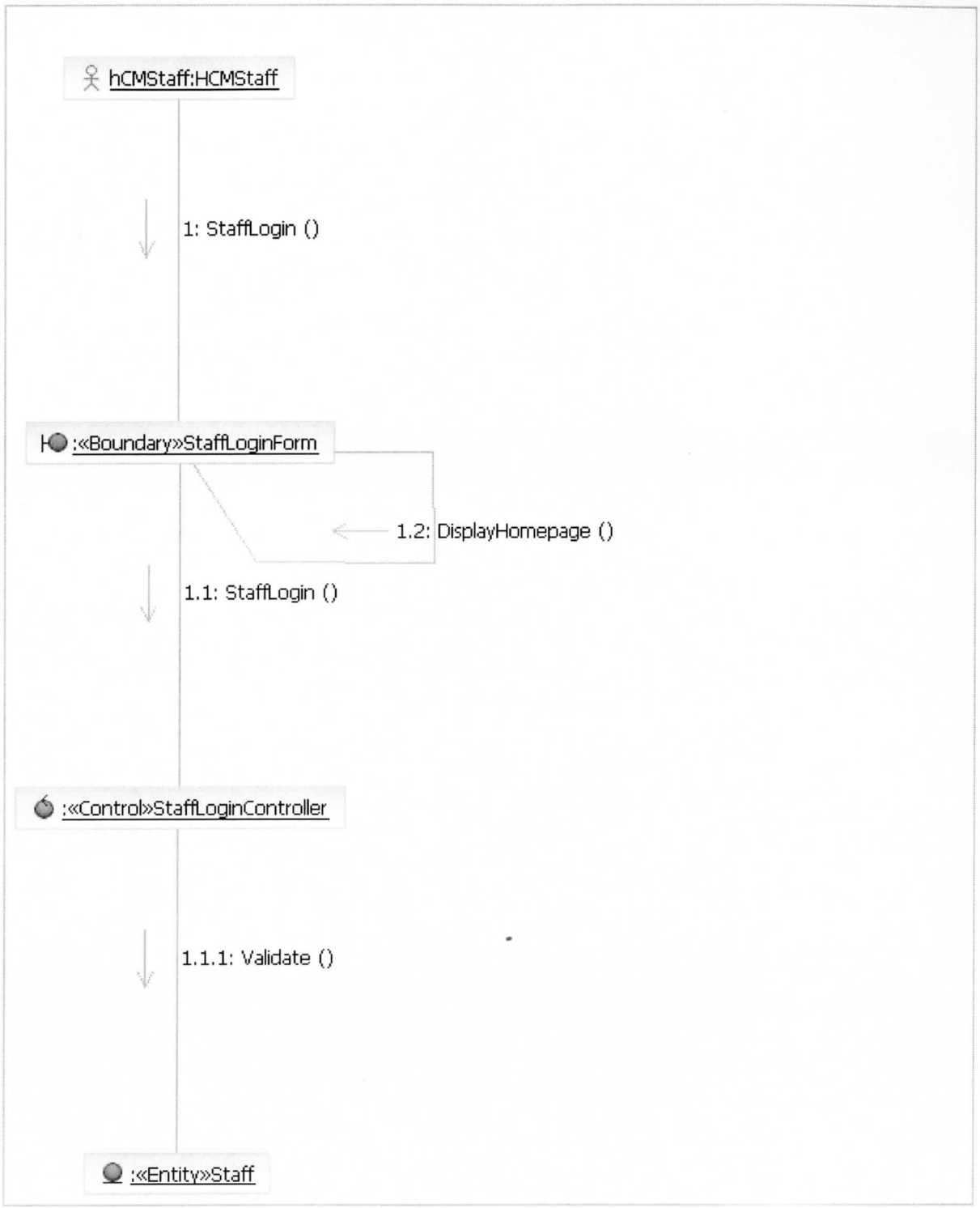


Figure 4-20 CD for Staff Login

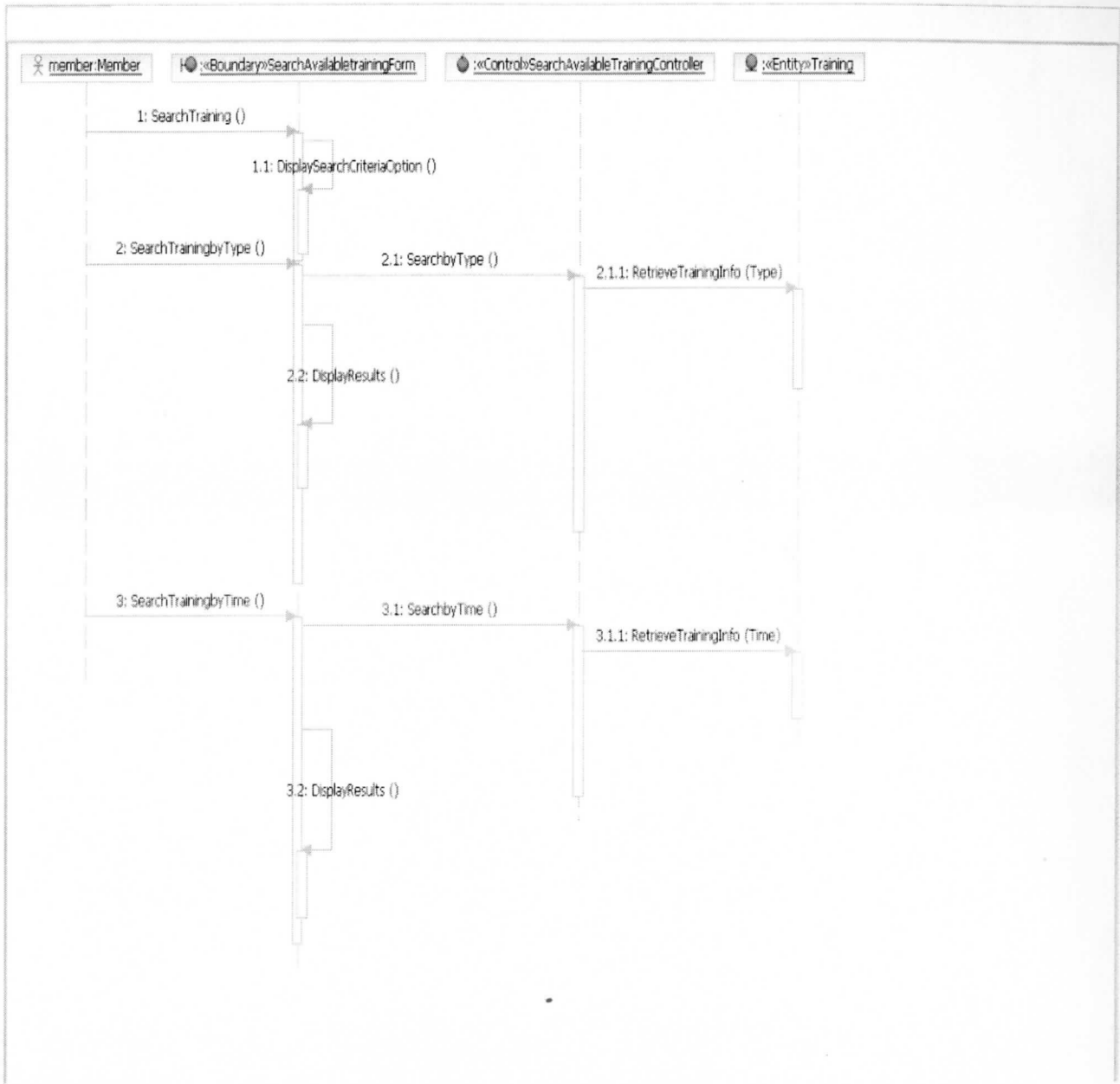


Figure 4-21 SD for Search Available Training

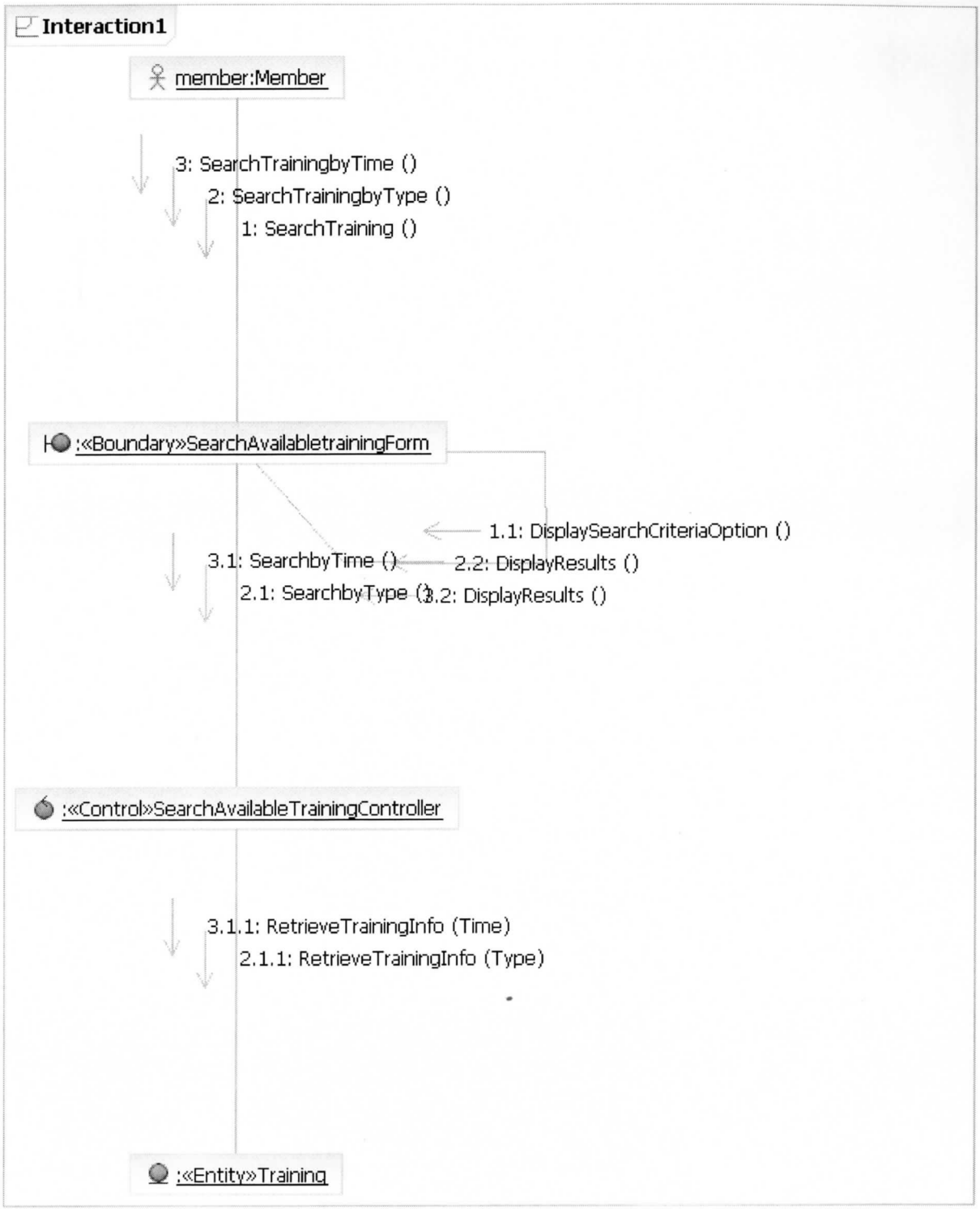


Figure 4-22 CD for Search Available Training

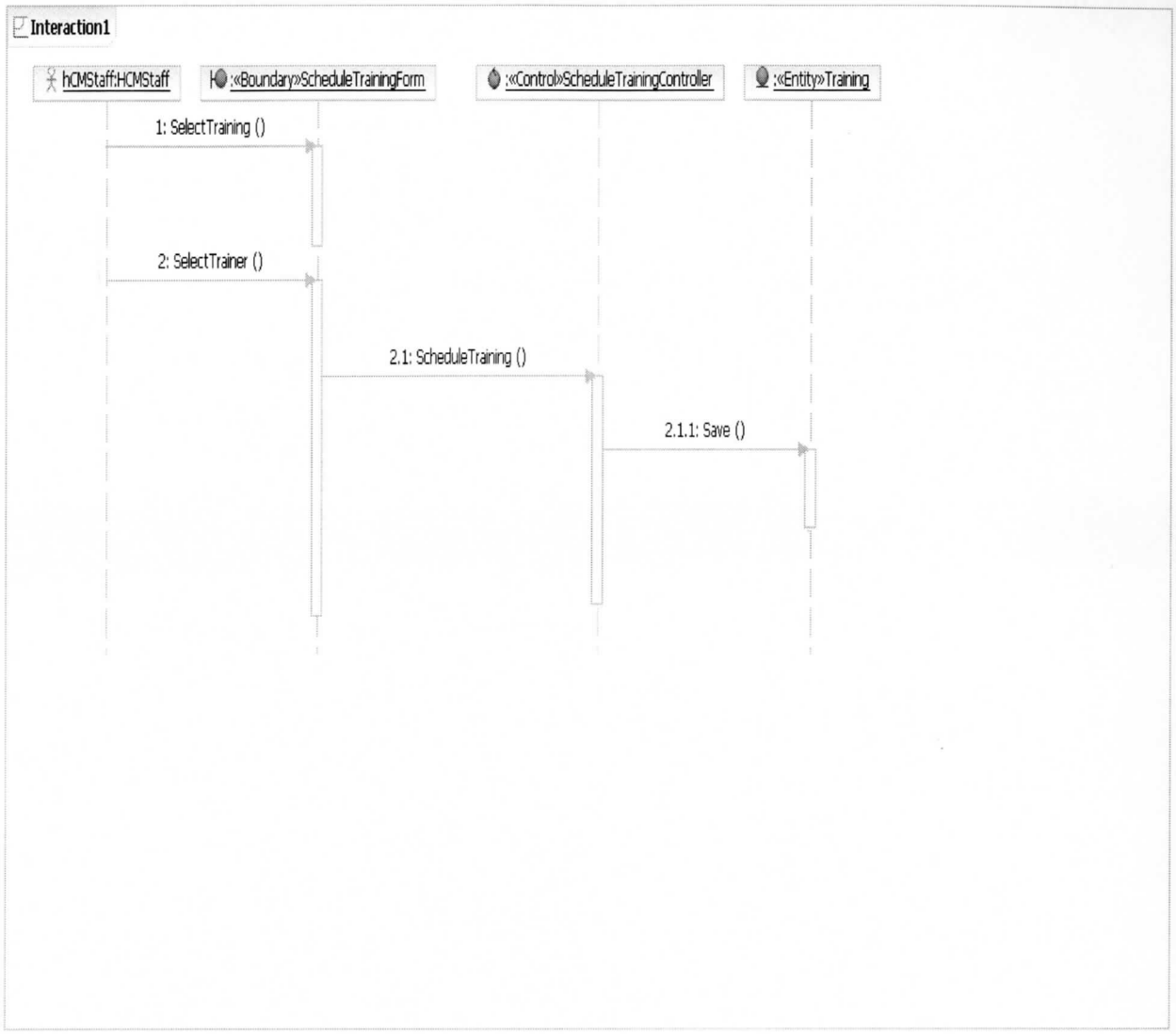


Figure 4-23 SD for Schedule Training

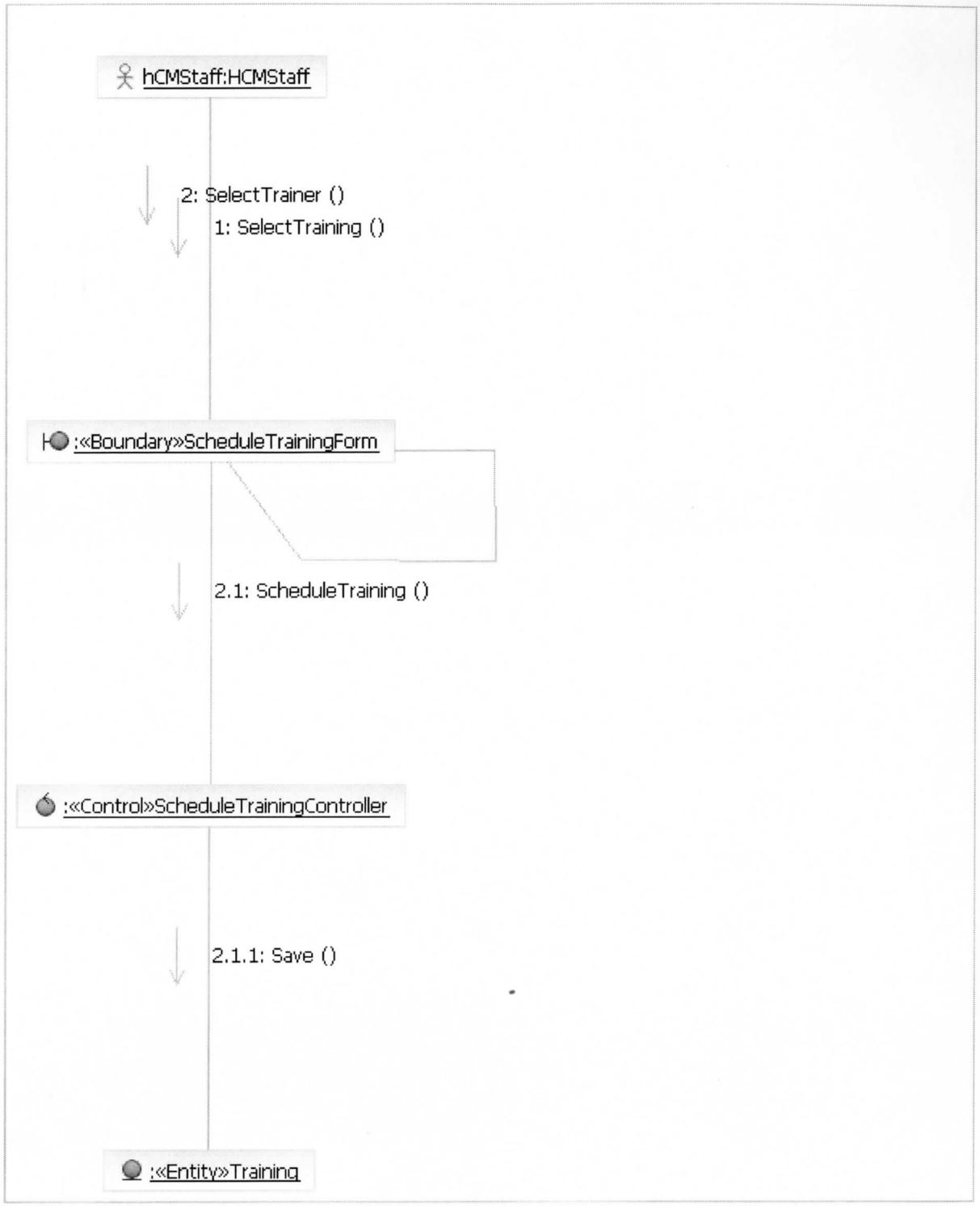


Figure 4-24 CD for Schedule Training

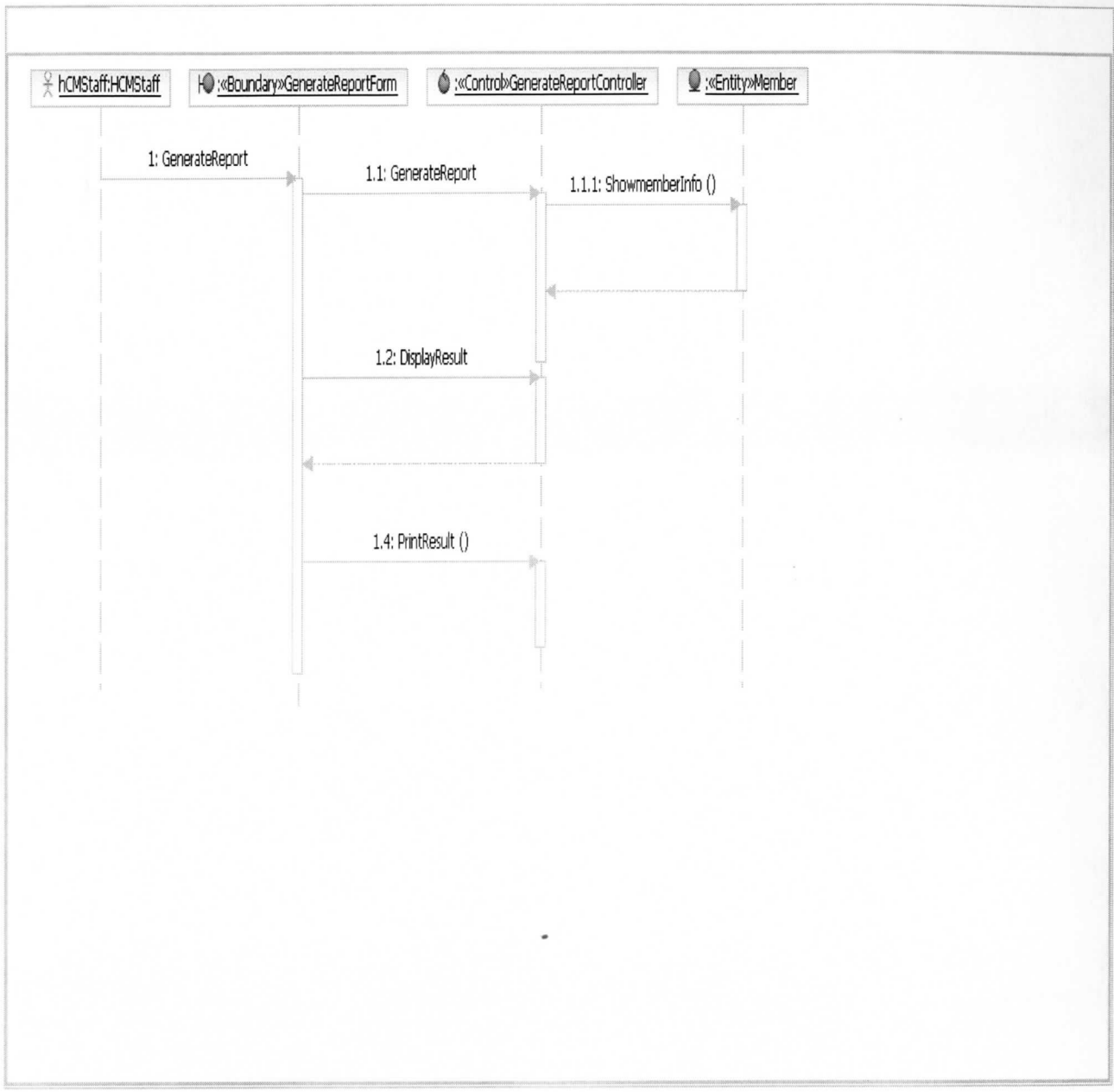


Figure 4-25 SD for Generate Reports

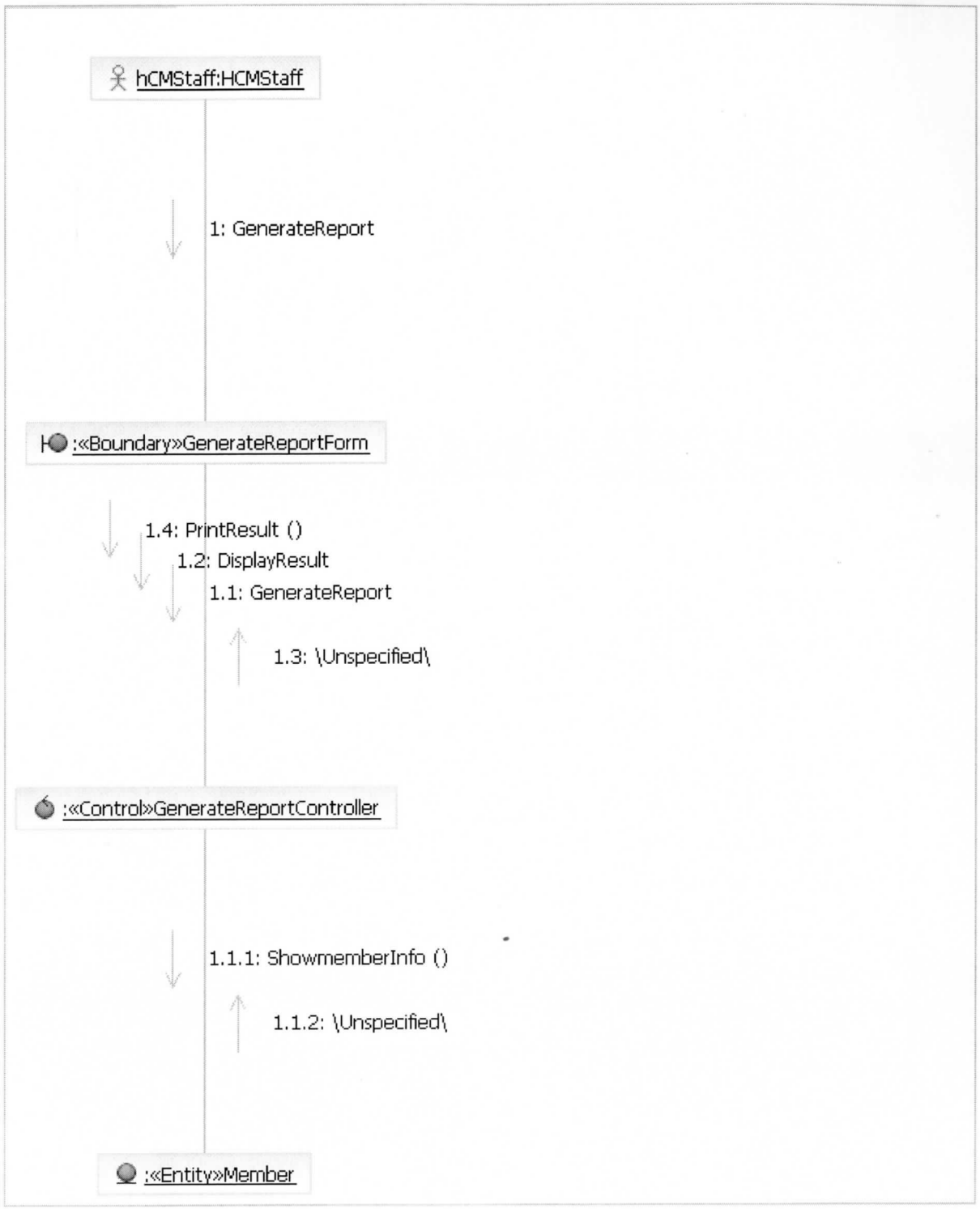


Figure 4-26 CD for Generate Reports

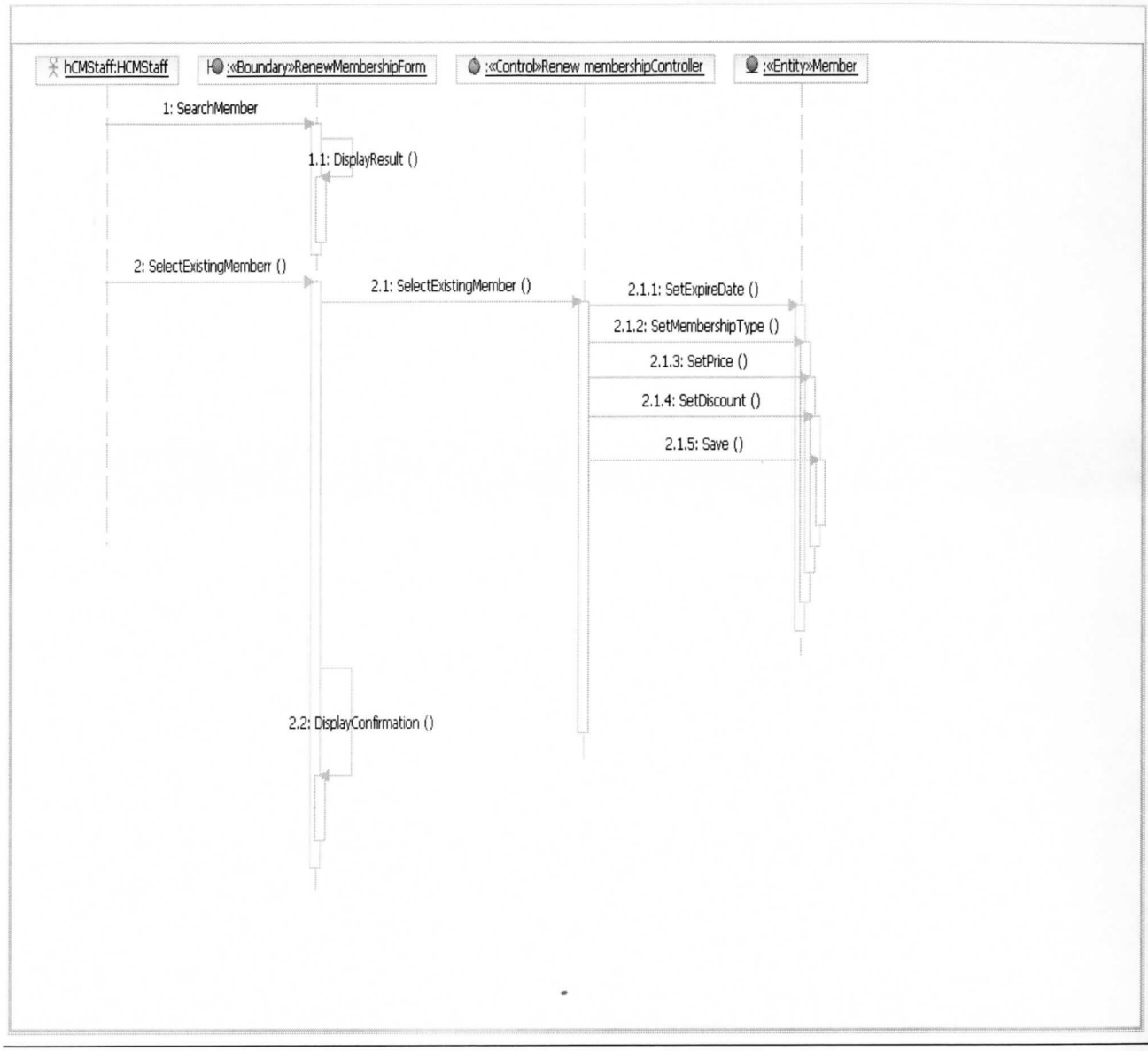


Figure 4-27 SD for Renew Membership

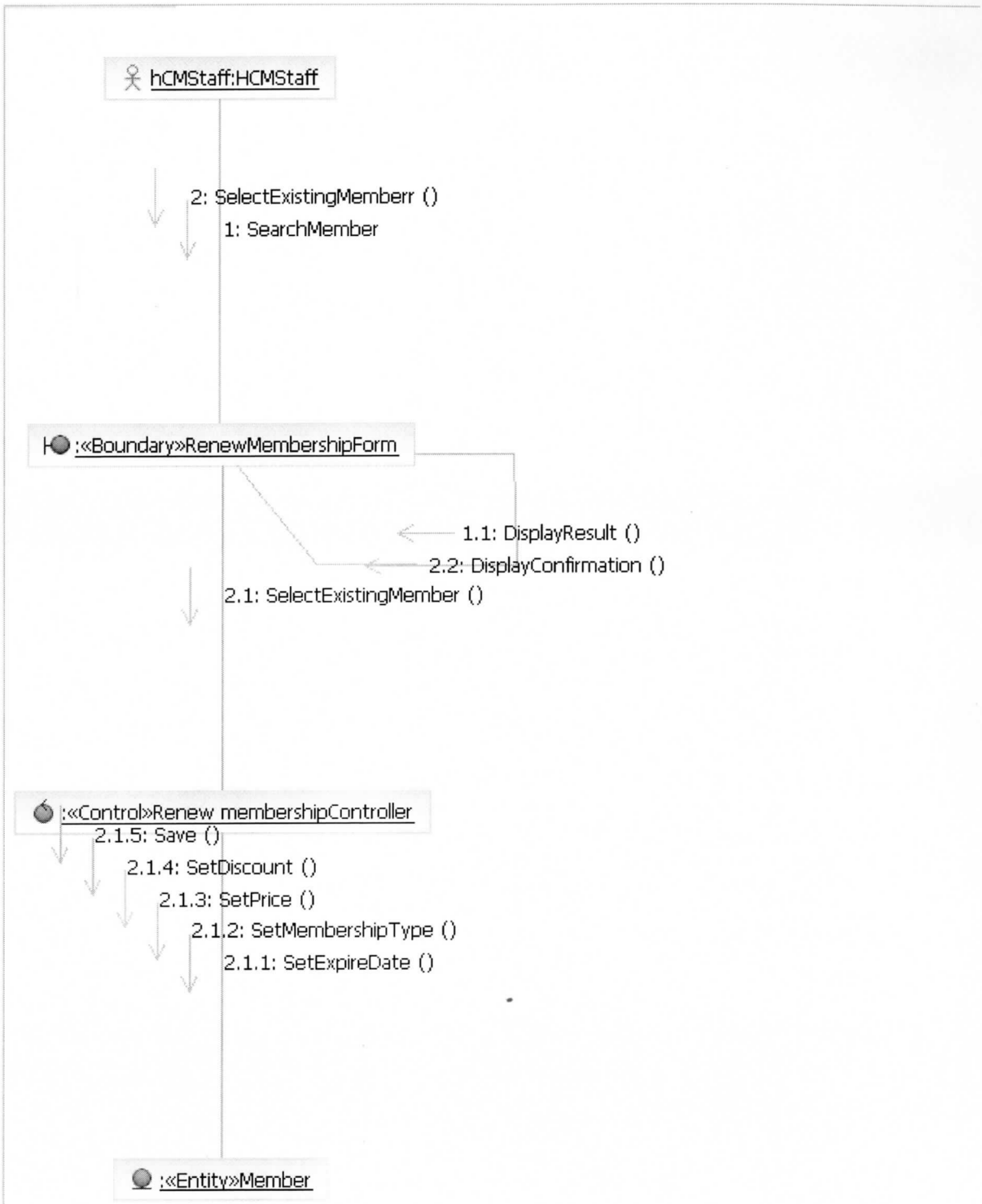


Figure 4-28 CD for Renew Membership

4.7 System Architecture

One of the important components of the system is the design of the hardware, network and software architecture. Systems architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user. In the case of HCMS, users will interact with the system through a web browser and will be able to interact from the comfort of their home, office and the Member service site. This will not only be used by clients and Members, but service technicians. The tables below describe in detail the hardware/software technical specifications needed to launch the system successfully, as well as some of the non-functional elements of the system. Systems architecture makes use of elements of both software and hardware and is used to enable design of such a composite system. Both are crucial to the success of the project and should be not overlooked.

HCM's System Requirements			
	Standard Client	Standard Webserver	Standard Database Server
Operating System	Windows XP or greater, OSX or Linux	Windows Enterprise Server	Windows Enterprise Server
Special Software	• 40 GB HD or more	• Apache 2.0 • Tomcat 6	• MySQL 5.5

	• Flash • Adobe Reader • Web Browser (Firefox 2+, IE 7+, Opera)	• Java 1.6	
Hardware	• Pentium 4 or better • 1 GB of Ram or more • 17 Inch Monitor • 40 GB HD or more • Keyboard & Mouse • 64 MB Video Card	• Dell PowerEdge M600 Blade Server • 2 Quad-Core Intel Xeon 5000 Sequence Processors • 16GB FDB DIMMs 667Mhz • 5x300GB Raid 5 (1.2 TB) • Offsite Tape Backup System • 32MB video card • Standard DVD-RW • Server Rack w/ Input Switch	• Dell PowerEdge M600 Blade Server • 2 Quad-Core Intel Xeon 5000 Sequence Processors • 16GB FDB DIMMs 667Mhz • 5x300GB Raid 5 (1.2 TB) • Offsite Tape Backup System • 32MB video card • Standard DVD-RW Server Rack w/ Input Switch

Network	• 10/100 Mb Ethernet connection • 1.5 Mbps+ DSL, Cable, Broadband Connection	• Dual OC3 Connection for redundancy • Dual Port Gb Ethernet	• Dual OC3 Connection for redundancy • Dual Port Gb Ethernet

Figure 4-29 System Requirements

Operational Requirements		
		The system will allow for users to browse the site via a web browser such as Internet Explorer or Firefox. Internet Explorer 6 is preferred.
		Members will only need a web browser as well as a internet connection.
		The system will write data to and from the database server
		The Member will update personal information from database
		The system will call a script to do nightly tape backups

		The system will need to keep up to web standards of speed, style, and dynamic functions.
		The database will need to be backed up (incrementally and full) onto tapes nightly

Performance Requirements		
		Response time needs to be as small as possible. No slower than 3 seconds per page load time
		The database will be updated real time, and backups won't affect performance
		Confirmation emails need to be sent under 30 sec
		The system should be able to accommodate up to 1000 simultaneous users at any given time without hurting browsing speeds.
		The database needs to be designed in a way to accommodate future growth.
		Being online, the system needs to have 24/7 run time and support
		The system should alert administrator's performance issues.

Security Requirements		

		Potentially could be worth around $100,000 for Member tracking data.
		Only staff will be given access to backend database to load new packages in or maintain user data.
		All Member data will be encrypted, as well as staff usernames and passwords.
		Virus protection will be installed on the system, along with new updates to the server to prevent security leaks.

Cultural Political Requirements		
		No special multilingual requirements are necessary
		No special customization requirements are necessary
		No special unstated requirements are necessary
		No special legal requirements are necessary

Figure 4-30 Non Functional Requirements

4.8 Data Storage Design

With the large amounts of data that HCM will have with their Member data base and continual growth through the years, they must have the capability to store massive

amounts of data as well as operate at desired performance expectations. HCM designed a data storage outline that will provide for both. A multidimensional database (MDB) is a type of database that is optimized for data warehouse and online analytical processing applications. HCM would benefit from using such a database due to their growing database and need for storage expansions and future developments. A multidimensional database implies that the ability is available to process the data in the database so that answers can be generated quickly. The development team will use MySQL for the database maintenance needs and any additional Member relations which will be easily identifiable and retrievable to pin point specific targets improving Member relations and the ability to further service the Member and their needs to the support services they need the most.

The indexed file solution will allow the database to have smaller files or tables but will make the database system faster and more robust. One of the disadvantages to this solution is that there are more tables; that will ultimately take up more space. HCM database administrators and the project team are prepared for their huge database of Members, their serviced items and the history; they have planned for over 30 years of data.

HCM will utilize Object Relational DBMS. This object-persistence format will take full advantage of the tools offered by MySQL, handle complex data types, and will be widely supported by the existing knowledge base of engineers.

4.9 Database Design:

Database design is one of the most crucial aspects of any modern application. The following E-R diagrams represent the logical database design of various modules in the application

Business Rules:

1. A CUSTOMER may get membership and renew membership.
2. SCHEDULE TRAINING must have one TRAINER and TRAINING.
3. A STAFF can schedule many TRAINING and TRAINING must schedule by a STAFF.
4. Each STAFF may serve many MEMBERs and a MEMBER is served by one STAFF.
5. Many STAFFs work in the HEALTH CLUB and a HEALTH CLUB has many STAFFs. A STAFF could be a manager or sales rep.
6. A STAFF can maintain many TRAINERs. Many TRAINERs are maintained by a STAFF.
7. A STAFF can maintain many MEMBERs. Many MEMBERs maintained by a STAFF.
8. A STAFF may accept many PAYMENTs. Many PAYMENTs are accepted by a STAFF.

Staff must enter relevant transaction information, such as payment type submitted by Members (e.g. cash or credit card), credit card number if credit card is used, payment amount, date of the payment, payment status, (confirm cash receipt or credit card approval).

Entity Types:

1. Name: STAFF
Type: Strong entity

Definition: People who are knowledgeable about the "Health Club Management" services, policies and procedures. They advise members about the use of the gym, handle money, and log daily reports.

Identifier: STAFF_ID

Attribute: STAFF_ID, Staff_Name, Staff_Address, Staff_User_Name, Staff_Password, Staff_Designation,

2. Name: MEMBER
Type: Strong entity

Definition: People in every age group that get a membership from "Health Club Management". Customers may have given membership that includes their name, address, phone number, and the expiration date of the membership.

Identifier: MEMBER_ID

Attribute: MEMBER_ID, Mem_Name, Mem_Address, Mem_Phone Number, Mem_E-mail, Mem_Member Type, Mem_Member Expiration, Mem_User_Name, Mem_Password.

3. Name: TRAINER
Type: Strong entity

Definition: Trainer helps to train the member. A staff schedules the trainer to training according to the requirements of the members.

Identifier: TRAINER_ID

Attribute: TRAINER_ID, Trainer_Name, Trainer_Address, Trainer_Phone, Trainer_User_Name, Trainer_Password, Trainer_Email.

4. Name: SCHEDULE TRAINING
Type: Strong entity

Definition: Schedule Training in an entity that combines trainer and training. It has trainer_ID, Training_ID, Staff, date , place, duration.

Identifier: SCHEDULE_ID

Attribute: SCHEDULE_ID, Sch_Tra_Date, Sch_Tra_Time, Sch_Tra_Place, TRAINING_ID, TRAINER_ID, STAFF_ID.

5. Name: TRAINING
Type: Strong entity

Definition: Training represents service, duration, type of the training and other information.

Identifier: TRAINING_ID

Attribute: TRAINING_ID, Training_Title, Training_Type, Training_Duration.

6. Name: PAYMENT
Type: Strong entity

Definition: It is a transfer of money between staff of the "Health Club Management" and the member (who taking the service) for the certain length of time. Payment could be in cash or credit card.

Identifier: PAY_ID

Attribute: PAY_ID, Pay_Amount, Pay_Date, Pay_Type, Pay_Status, MEMBER_ID, Credit_Card_Number.

7. Name: CREDIT CARD
Type: Strong entity

Definition: Detailed credit card payment information.

Identifier: CREDIT_CARD_NUMBER

Attribute: CREDIT_CARD_NUMBER, CR_Card_Name, CR_Card_Address, CR_Card_Type, CR_Card_Expiration_Date, MEMBER_ID.

Relationship Types:

1. Name: Staff maintain member
Type: Binary: 1:M

Description: The staff who maintain the member.

2. Name: Payment accepted by Staff
Type: M:1

Description: payment processed by staff.

3. Name: Payment paid by member.
Type: M:1

Description: associates payment and staff, expresses the unique staff who accepted the payment.

4. Name: Credit Card used in payment
Type: M:M

Description: associates member to payment, convey the payment record for the member.

5. Name: Trainer included in schedule training.
Type: 1:M

Description: associates trainer to schedule training.

6. Name: Training included in schedule training.
Type: 1:M

Description: associates training to schedule training.

7. Name: Scheduled Training by staff
Type: M:1

Description: associates staff to schedule training.

ER Diagram-HCM System

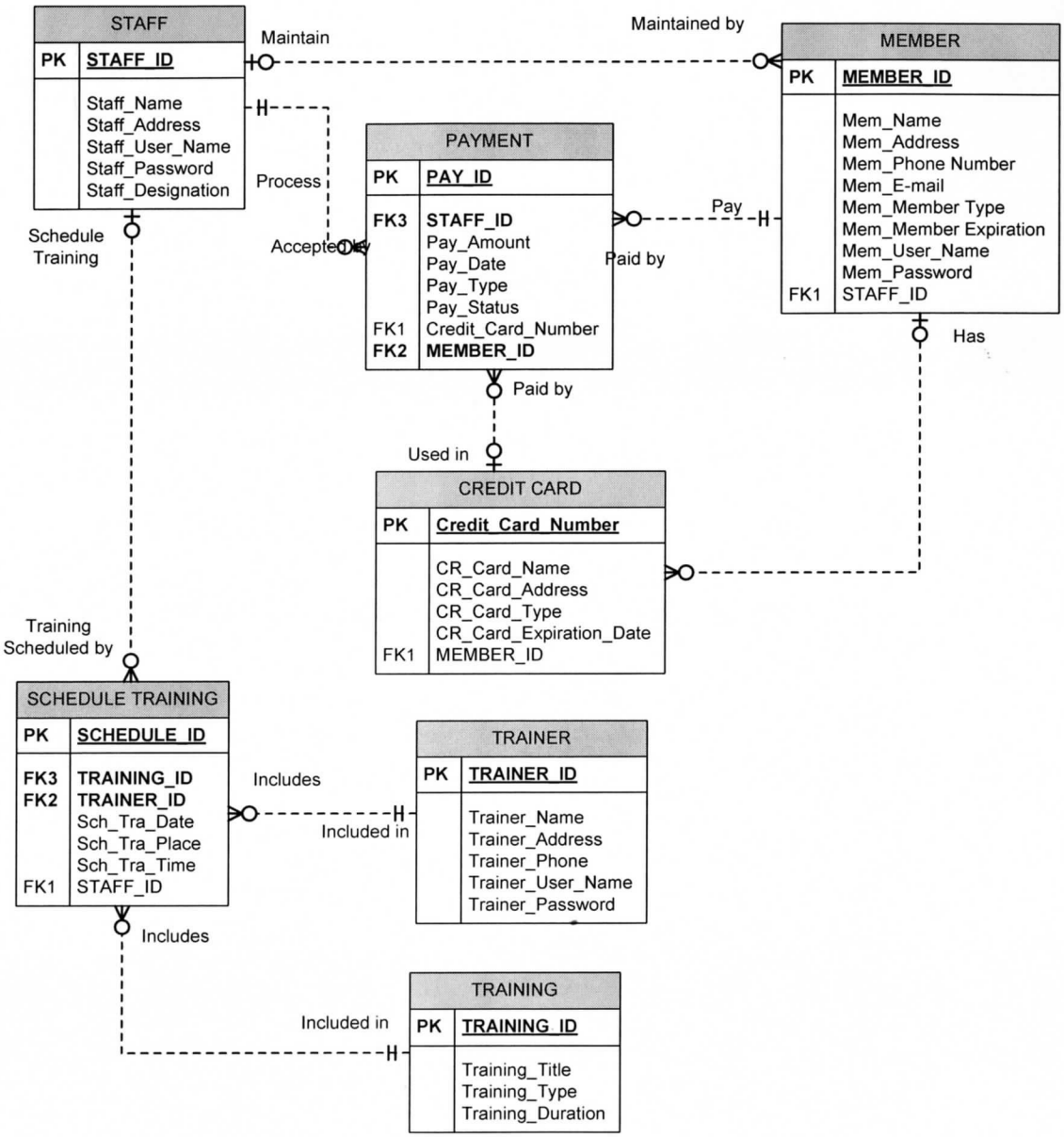


Figure 4-31 ER Diagram

Data Dictionary:

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	REQUIRED	PK OR FK	FK REFERENCE TABLE
STAFF	STAFF_ID	Staff ID	NUMERIC(10)	Y	PK	
	Staff_Name	Staff Name	VARCHAR(25)	Y		
	Staff_Address	Staff Address	VARCHAR(50)	Y		
	Staff_User_Name	Staff User Name	VARCHAR(15)	Y		
	Staff_Password	Staff Password	VARCHAR(15)	Y		
	Staff_Designation	Staff Designation	VARCHAR(10)	Y		
MEMBER	MEMBER_ID	MEMBER_ID	NUMERIC(10)	Y	PK	
	Mem_Name	Member Name	VARCHAR(25)	Y		
	Mem_Address	Member Address	VARCHR(50)	Y		
	Mem_Phone Number	Mem_Phone Number	VARCHAR(15)	Y		
	Mem_E-mail	Member E-mail	VARCHAR(20)	Y		
	Mem_Member Type	Member Type	VARCHAR(15)	Y		
	Mem_Member Expiration	Member Expiration	DATE			
TRAINER	TRAINER_ID	TRAINER_ID	NUMERIC(10)	Y	PK	
	Trainer_Name	Trainer Name	VARCHR(25)	Y		
	Trainer_Address	Trainer Address	VARCHR(50)	Y		
	Trainer_Phone	Trainer Phone	VARCHAR(15)	Y		
	Trainer_User_Name	Trainer User Name	VARCHAR(15)	Y		
	Trainer_Password	Trainer Password	VARCHAR(15)			
	Trainer_Email	Trainer Email	VARCHAR(20)			

SCHEDULE TRAINING	SCHEDULE_ID	Schedule Training ID	NUMERIC(10)	Y	PK	
	Sch_Tra_Date	Schedule Training Date	DATE	Y		
	Sch_Tra_Place	Schedule Training Place	VARCHAR(20)	Y		
	Sch_Tra_Time	Schedule Training Time	DATE	Y		
	TRAINING_ID	TRAINING ID	NUMERIC(10)	Y	FK	TRAINING
	TRAINER_ID	TRAINER_ID	NUMERIC(10)	Y	FK	TRAINER
	STAFF_ID	STAFF_ID	NUMERIC(10)		FK	STAFF
TRAINING	TRAINER_ID	Training ID	NUMERIC(10)	Y	PK	
	Training_Title	Training Title	VARCHR(20)	Y		
	Training_Type	Training Type	VARCHR(15)	Y		
	Training_Duration	Training Duration	VARCHR(20)	Y		
PAYMENT	PAY_ID	Payment ID	NUMERIC(10)	Y	PK	
	Pay_Amount	Payment Amount	NUMERIC(10)	Y		
	Pay_Status	Payment Status	VARCHAR(20)	Y		
	Pay_Date	Payment Date	DATE	Y		
	Pay_Type	Payment Type	VARCHAR(15)	Y		
	MEMBER_ID	Member ID	NUMERIC(10)	Y	FK	MEMBER
	Credit_Card_Number	Credit Card Number	VARCHAR(15)		FK	CREDITCARD

CREDIT CARD	Credit_Card_Number	Credit Card Number	VARCHAR(15)	Y	PK	
	CR_Card_Name	Credit Card Name	VARCHAR(15)	Y		
	CR_Card_Address	Credit Card Address	VARCHAR(50)	Y		
	CR_Card_Type	Credit Card Type	VARCHAR(15)	Y		
	CR_Expiration_Date	Credit Expiration Date	DATE	Y		
	MEMBER_ID	MEMBER_ID	NUMERIC(10)	Y	FK	MEMBER

HCM Database Script:

Database: MySQL
Version: 5.5
Database Name: hcmsdb
User: root
Password: admin

```
drop table STAFF;
drop table MEMBER;
drop table TRAINER;
drop table TRAINING;
drop table SCHEDULETRAINING;
drop table CREDITCARD;
drop table PAYMENT;
drop table CONTACT;


CREATE TABLE STAFF (
STAFF_ID                INTEGER UNSIGNED   NOT NULL
AUTO_INCREMENT,
Staff_Name                VARCHAR(25)              not null,
Staff_Address             VARCHAR(50)              not null,
Staff_User_Name            VARCHAR(15)             not null,
Staff_Password            VARCHAR(15)              not null,
Staff_Designation         VARCHAR(10)              not null,
```

```
PRIMARY KEY (STAFF_ID)
)ENGINE=InnoDB;


insert into STAFF
(Staff_Name,Staff_Address,Staff_User_Name,Staff_Password,Staff_Designation)
values ('Muhammad','432 Bellevue Way SE','Rahim','8765','Manager');

insert into STAFF
(Staff_Name,Staff_Address,Staff_User_Name,Staff_Password,Staff_Designation)
values ('Ahammad','432 Bellevue Way SE','Rahim','8765','Manager');




CREATE TABLE MEMBER (
MEMBER_ID                INTEGER UNSIGNED      NOT NULL
AUTO_INCREMENT,
Mem_Name                 VARCHAR(25)           not null,
Mem_Address              VARCHAR(50)           not null,
Mem_Phone_Number         VARCHAR(15)           not null,
Mem_Email                VARCHAR(20)           not null,
Mem_Member_Type          VARCHAR(15)           not null,
mem_Member_StartDate     DATE                  not null,
Mem_Member_Expiration    DATE                  not null,
Mem_User_Name            VARCHAR(10)           not null,
Mem_Password             VARCHAR(15)           not null,
PRIMARY KEY (MEMBER_ID)
)ENGINE=InnoDB;


insert into MEMBER
(Mem_Name,Mem_Address,Mem_Phone_Number,Mem_Email,Mem_Member_Type,m
em_Member_StartDate,Mem_Member_Expiration,Mem_User_Name,Mem_Password)
values ('Muhammad','432 Bellevue Way
SE','7656984356','bht@yahoo.com','Clerck','2009-5-23','2005-3-
23','muhammad','password');


CREATE TABLE TRAINER (
TRAINER_ID               INTEGER UNSIGNED      NOT NULL
AUTO_INCREMENT,
Trainer_Name             VARCHAR(25)           not null,
```

```
Trainer_Address          VARCHAR(50)          not null,
Trainer_Phone            VARCHAR(15)          not null,
Trainer_User_Name        VARCHAR(15)          not null,
Trainer_Password         VARCHAR(15)          not null,
Trainer_Email            VARCHAR(20)          not null,
PRIMARY KEY (TRAINER_ID)
)ENGINE=InnoDB;

insert into TRAINER
(Trainer_Name,Trainer_Address,Trainer_Phone,Trainer_User_Name,Trainer_Password,
Trainer_Email)
values ('Muhammad','432 Bellevue Way
SE','7656984356','Rahim','876','bht@yahoo.com');


CREATE TABLE TRAINING (
TRAINING_ID              INTEGER UNSIGNED     NOT NULL
AUTO_INCREMENT,
Training_Title       VARCHAR(20)              not null,
Training_Type        VARCHAR(15)              not null,
Training_Duration    VARCHAR(20)              not null,
PRIMARY KEY (TRAINING_ID)
)ENGINE=InnoDB;

insert into TRAINING
(Training_Title,Training_Type,Training_Duration)
values ('Muhammad','Walking','2 hour');


CREATE TABLE SCHEDULETRAINING (
SCHEDULE_ID              INTEGER UNSIGNED     NOT NULL
AUTO_INCREMENT,
Sch_Tra_Date             DATE                 not null,
Sch_Tra_Place            VARCHAR(20)          not null,
Sch_Tra_Time             TIME                 not null,
TRAINING                 VARCHAR(20)          not null,
TRAINER                  VARCHAR(20)          not null,
STAFF                    VARCHAR(20)          not null,
PRIMARY KEY (SCHEDULE_ID)
)ENGINE=InnoDB;

insert into SCHEDULETRAINING
(Sch_Tra_Date,Sch_Tra_Place,Sch_Tra_Time,TRAINING,TRAINER,STAFF)
```

```
values ('2011-3-3','432 Bellevue Way SE','11:20:00','1--body building','1--muhammad','1--muhammad');


CREATE TABLE CREDITCARD
(Credit_Card_Number                VARCHAR(15)              PRIMARY
KEY,
CR_Card_Name                       VARCHAR(15)              not null,
CR_Card_Address                    VARCHAR(50)              not null,
CR_Card_Type                       VARCHAR(15)              not null,
CR_Expiration_Date          DATE                  not null,
MEMBER_ID                   INTEGER UNSIGNED      not null,
FOREIGN KEY (MEMBER_ID) REFERENCES MEMBER (MEMBER_ID)
)ENGINE=InnoDB;


insert into CREDITCARD
(Credit_Card_Number,CR_Card_Name,CR_Card_Address,CR_Card_Type,CR_Expirati
on_Date,MEMBER_ID)
values ('1','Nargis','432 Bellevue Way SE','Visa','2012-09-12','1');



CREATE TABLE PAYMENT
(PAY_ID                    MEDIUMINT              NOT NULL
AUTO_INCREMENT,
Pay_Amount          NUMERIC(10)                   not null,
Pay_Status          VARCHAR(20)                   not null,
Pay_Date            DATE                          not null,
Pay_Type            VARCHAR(15)                   not null,
MEMBER_ID           INTEGER UNSIGNED              not null,
Credit_Card_Number VARCHAR(15),
PRIMARY KEY (PAY_ID),
FOREIGN KEY (MEMBER_ID) REFERENCES MEMBER (MEMBER_ID),
FOREIGN KEY (Credit_Card_Number) REFERENCES CREDITCARD
(Credit_Card_Number)
)ENGINE=InnoDB;

insert into PAYMENT
(PAY_ID,Pay_Amount,Pay_Status,Pay_Date,Pay_Type,MEMBER_ID,Credit_Card_Nu
mber)
values ('1','564','Paid','2012-09-12','Credit card','1','1');


CREATE TABLE CONTACT (
```

```
CONTACT_ID              INTEGER UNSIGNED      NOT NULL
AUTO_INCREMENT,
Contact_Name             VARCHAR(25)          not null,
Contact_Phone           VARCHAR(50)           not null,
Contact_Email           VARCHAR(15)           not null,
Contact_Comment         VARCHAR(15)           not null,
PRIMARY KEY (CONTACT_ID)
)ENGINE=InnoDB;
```

CHAPTER 5

PROJECT IMPLEMENTATION

This section captures all the user interfaces of the system. This includes visitor panel, member panel, trainer panel, and admin panel. Visitor panel for the external user and how they see the system from the out side of the company. Member panel help the member to login to the system and view their information. Trainer panel is for the trainer to login to the system and view their information. Admin panel is for the admin of the system. Admin can login to the system from this panel to add user, trainer, and training, schedule training, and generate report.

HCMS Home Page: This is the home page of newly developed system.

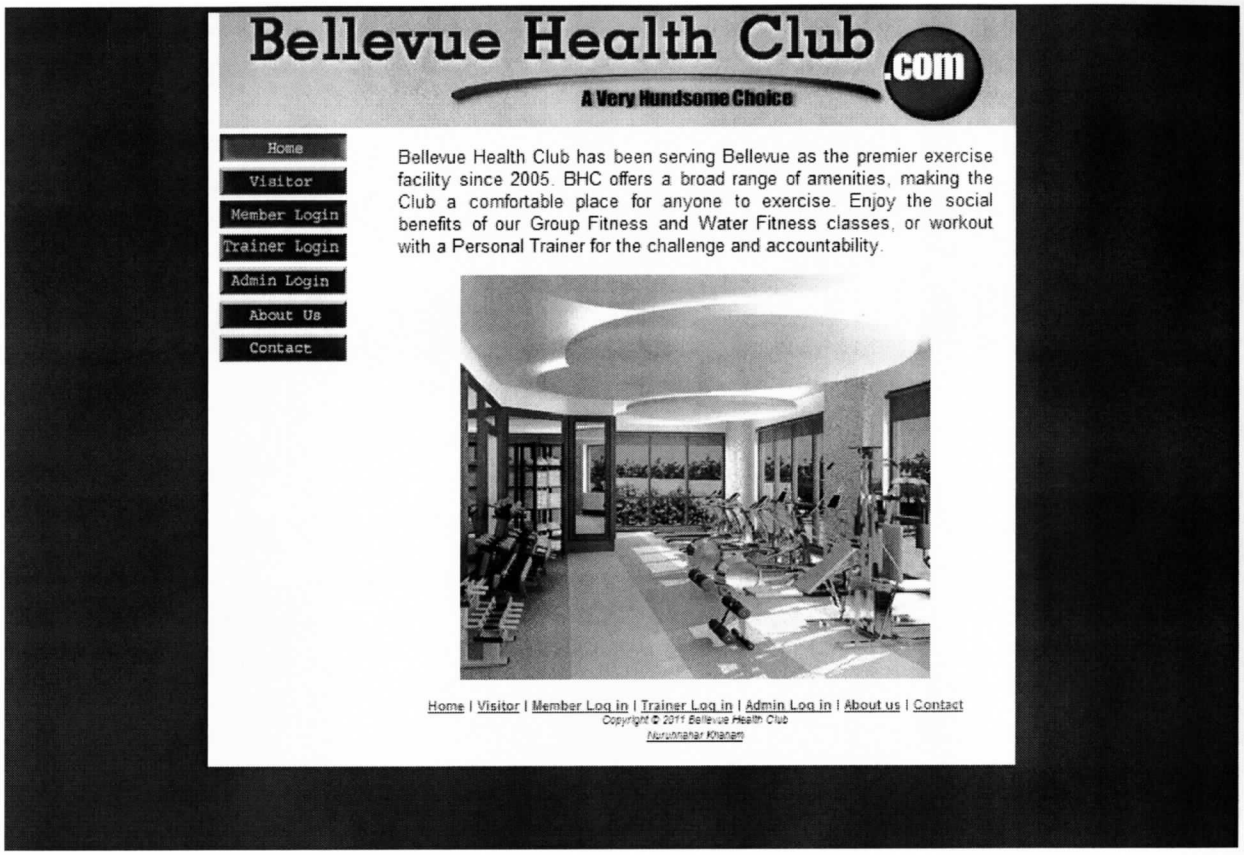


Figure 5-1 Home Page

Visitor Panel:

Visitor Home Page:

Visitor panel for the external user and how they see the system from the out side of the company.

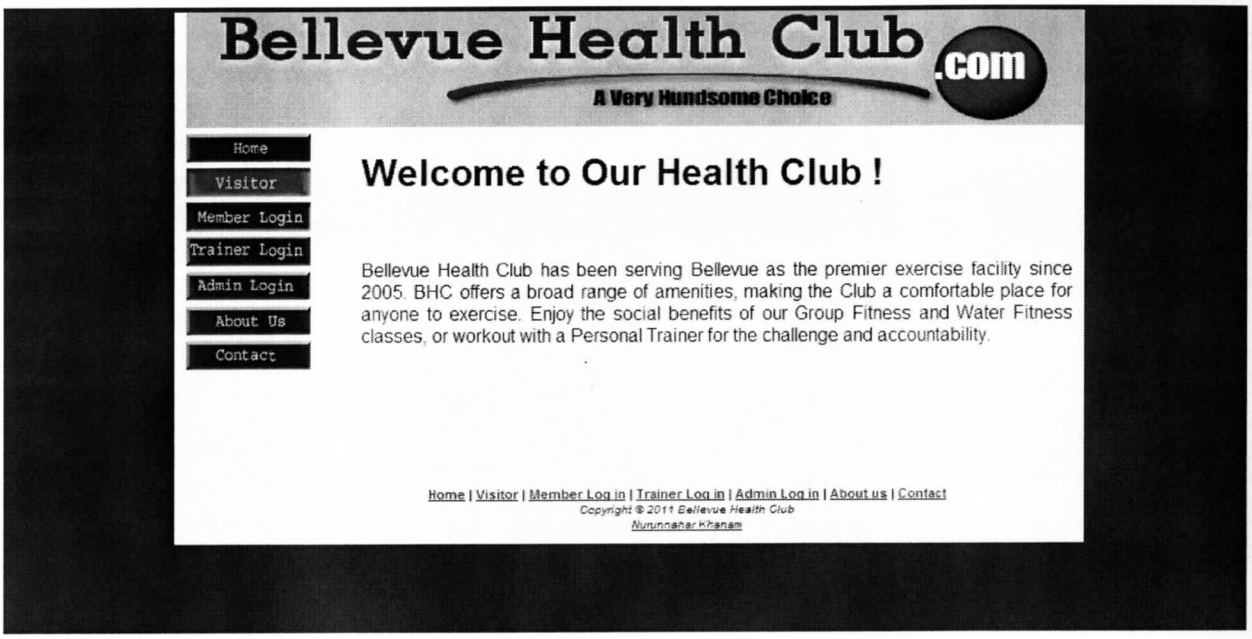


Figure 5-2 Visitor Page

Contact Page:

Visitors can contact from this page.

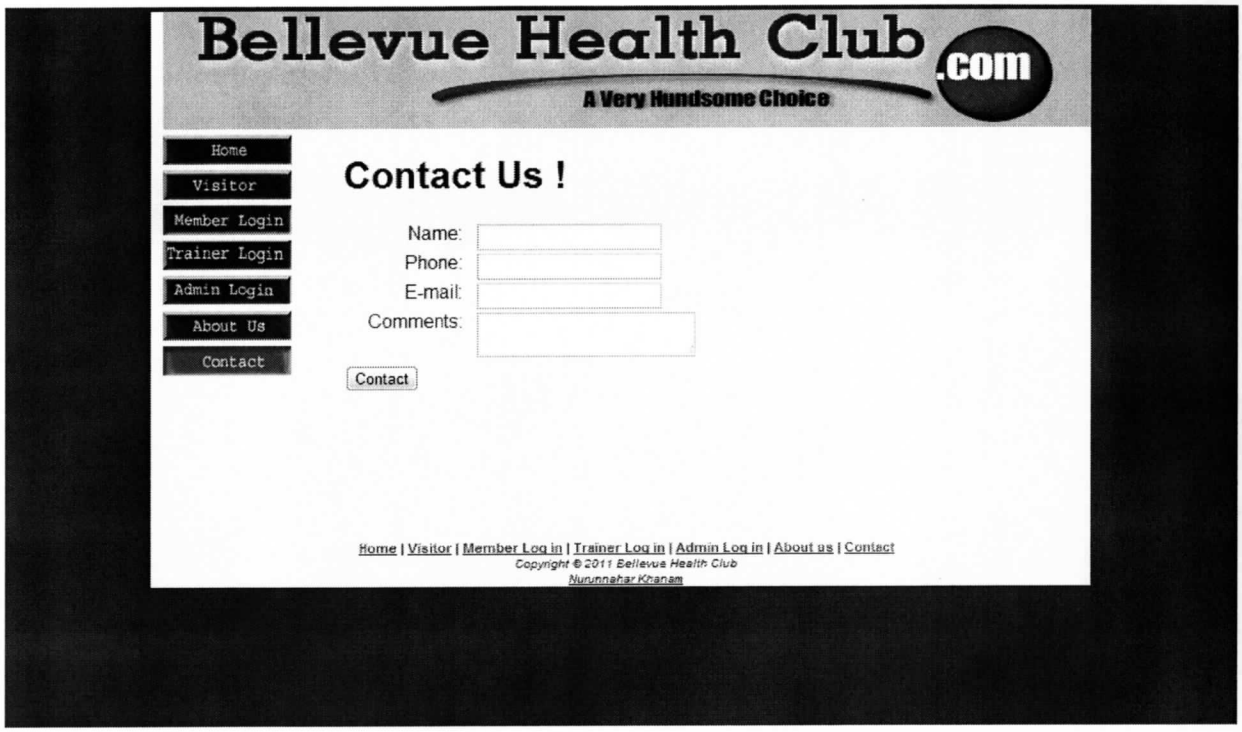


Figure 5-3 Contact Page

About Us Page:

Visitors can know about the HCMS from this page.

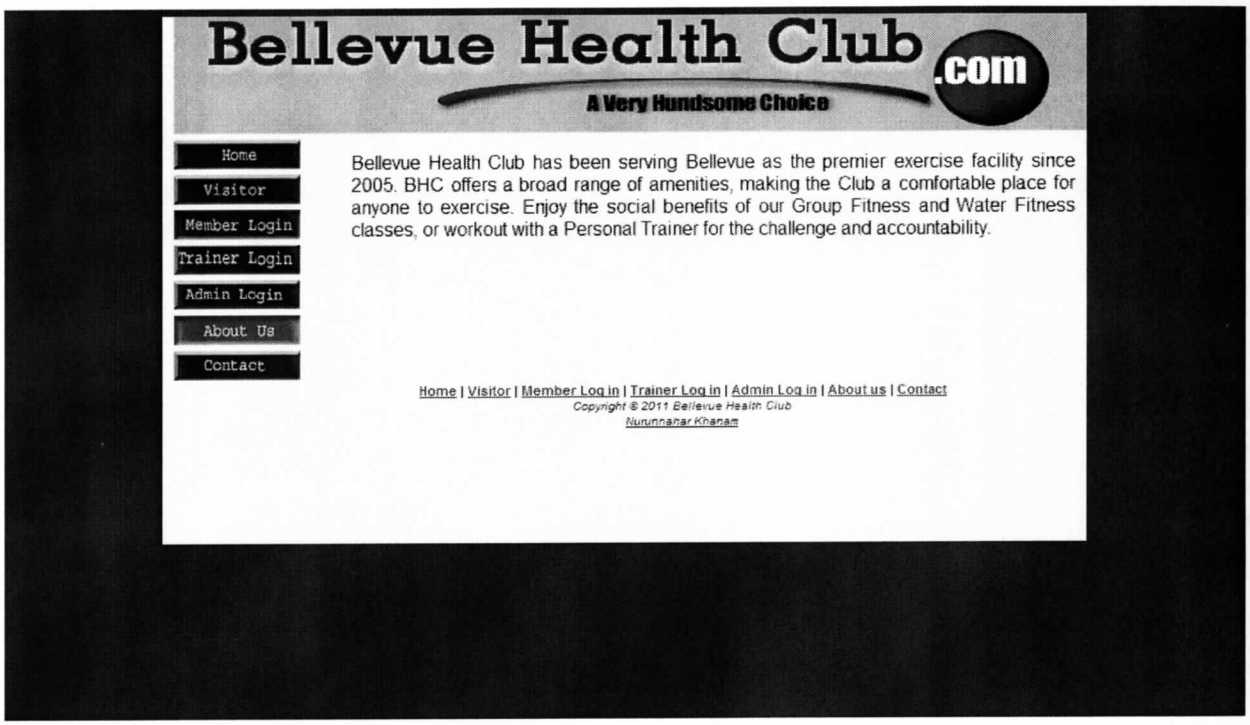


Figure 5-4 About Us Page

Member Panel:

Member panel help the member to login to the system and view their information.

Login Page: Member can go to login page by clicking the "Member Login" button of system home page. Then member can login to the system from this page. Member need to insert user name and password in the regarding field and then click in "Log in" button to go to the member home page.

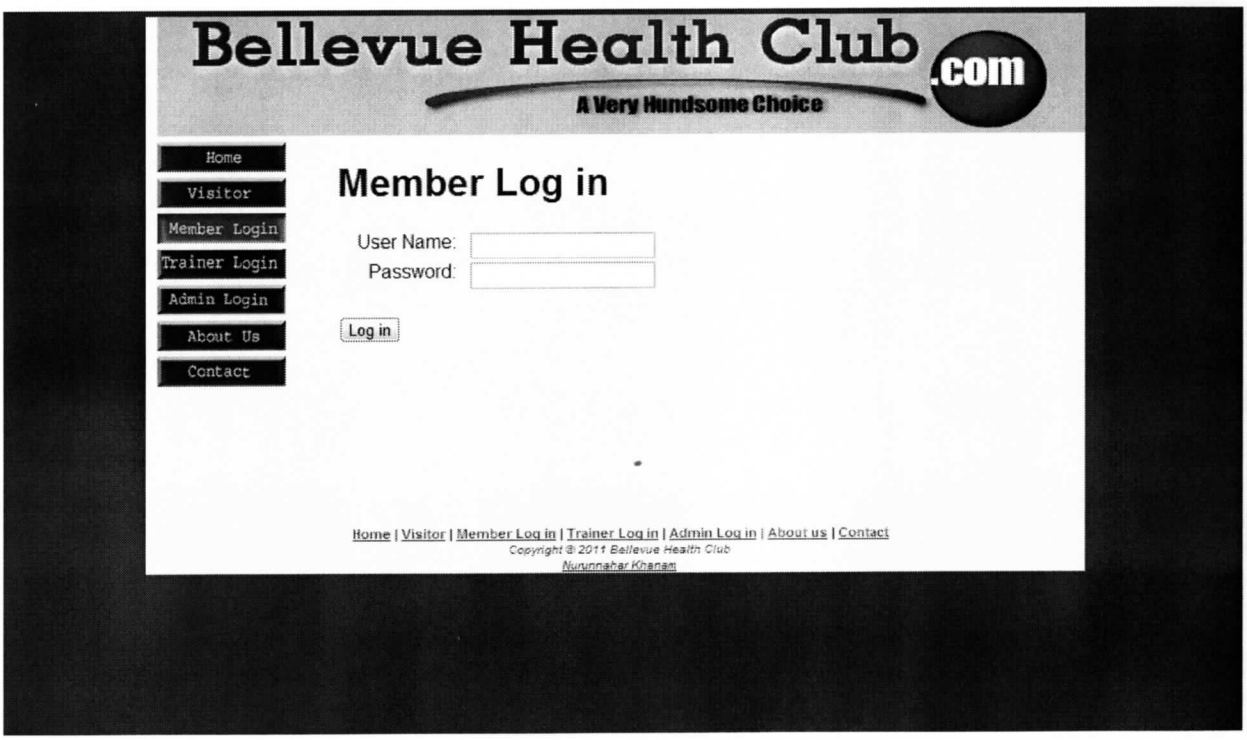


Figure 5-5 Member Login Page

Member Home Page: Member can search available training session from the member home page.

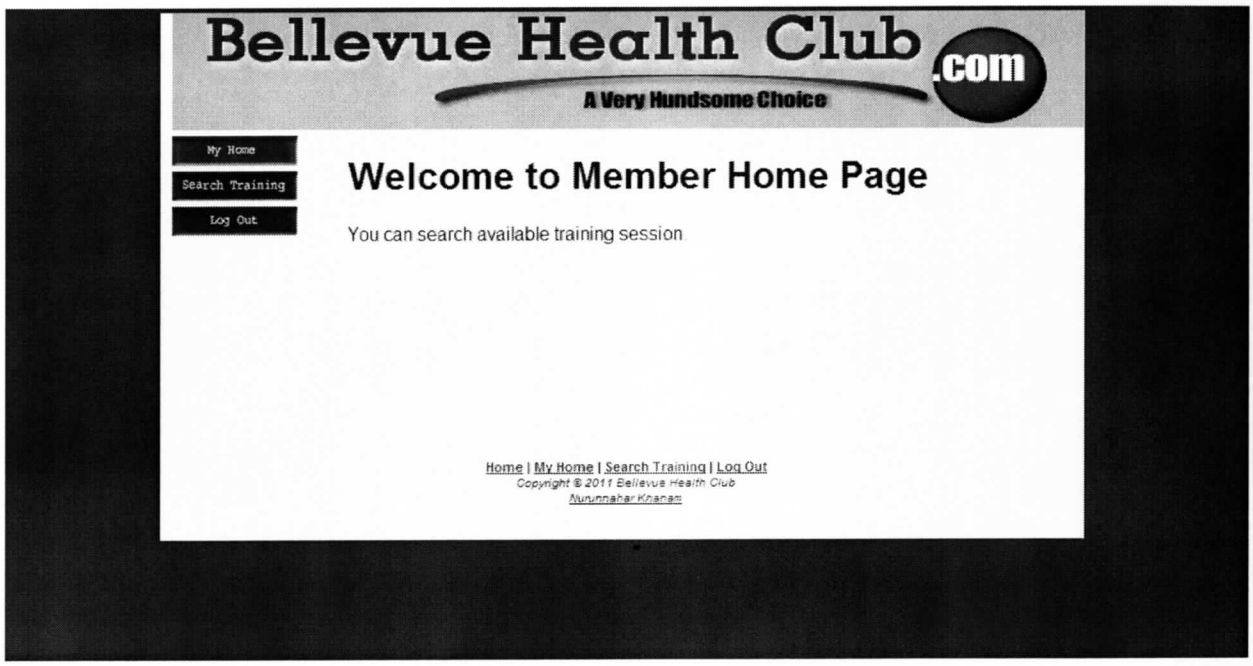


Figure 5-6 Member Home Page

Search Training Page:

Member can search training from the page. Member can go to search training page by clicking the "Search Training" button. Then member need to click "Search" button after selecting training place and insert training date and time.

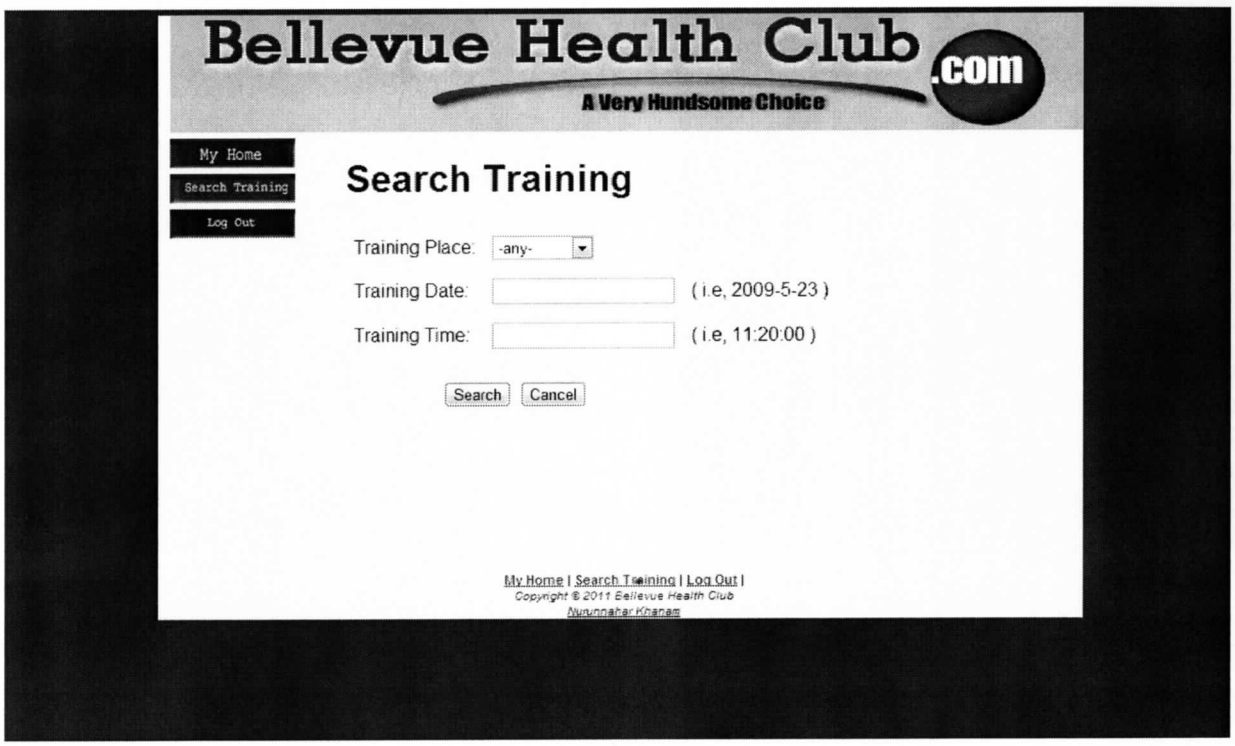


Figure 5-7 Search Training Page

Search Result:

Member can see the search result here.

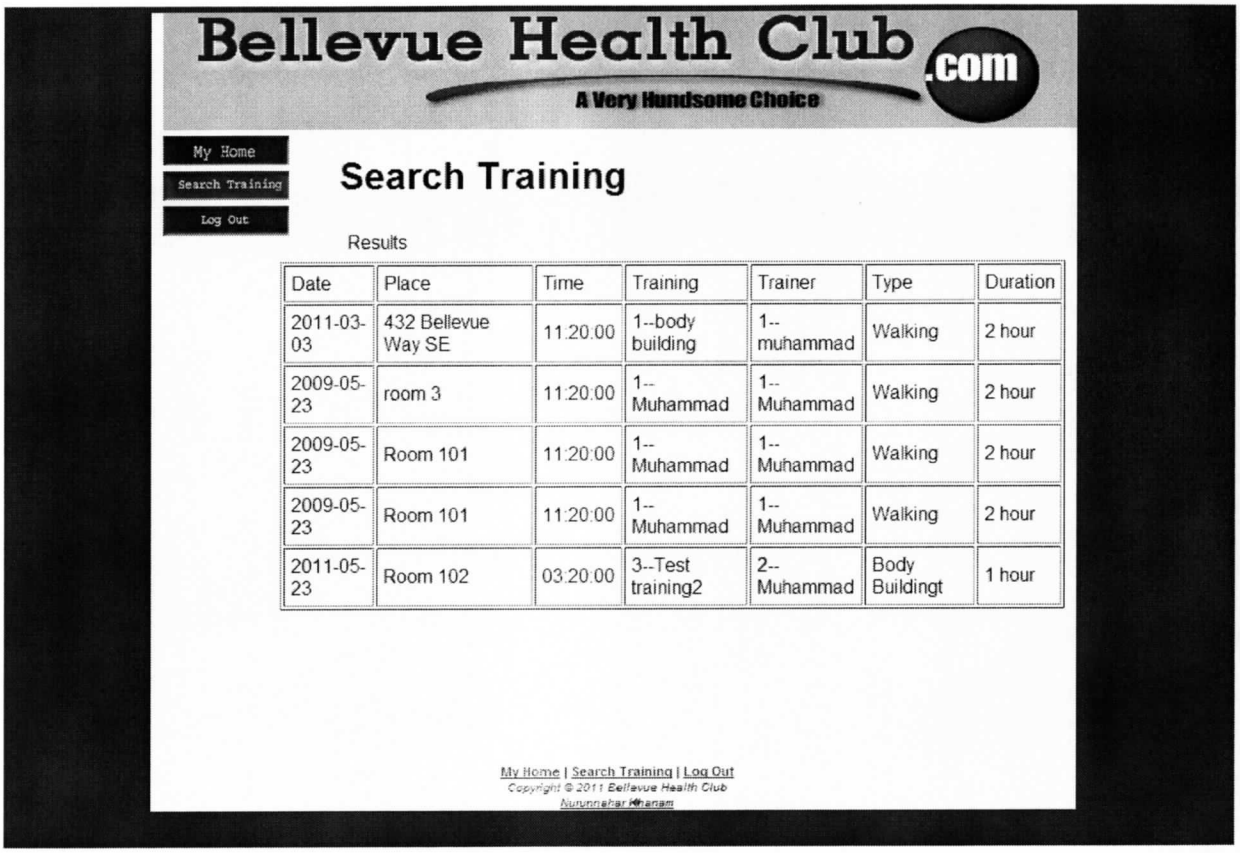


Figure 5-8 Search Result Page

Trainer Panel:

Trainer Login: trainer can go to Trainer Log in page by clicking the "Trainer Log in" button. Trainer need to insert user name and password in the regarding field and then click in "Log in" button to go to the trainer home page.

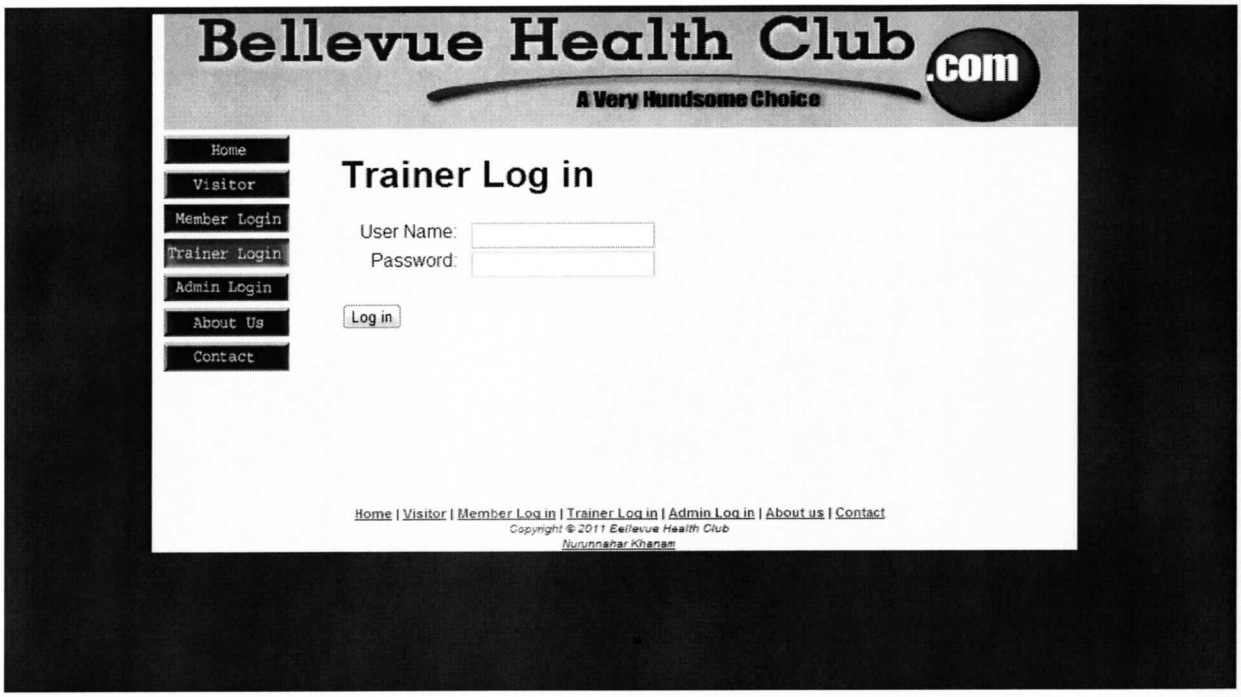


Figure 5-9 Trainer Login Page

Trainer Home Page: Trainer can search training from the page:

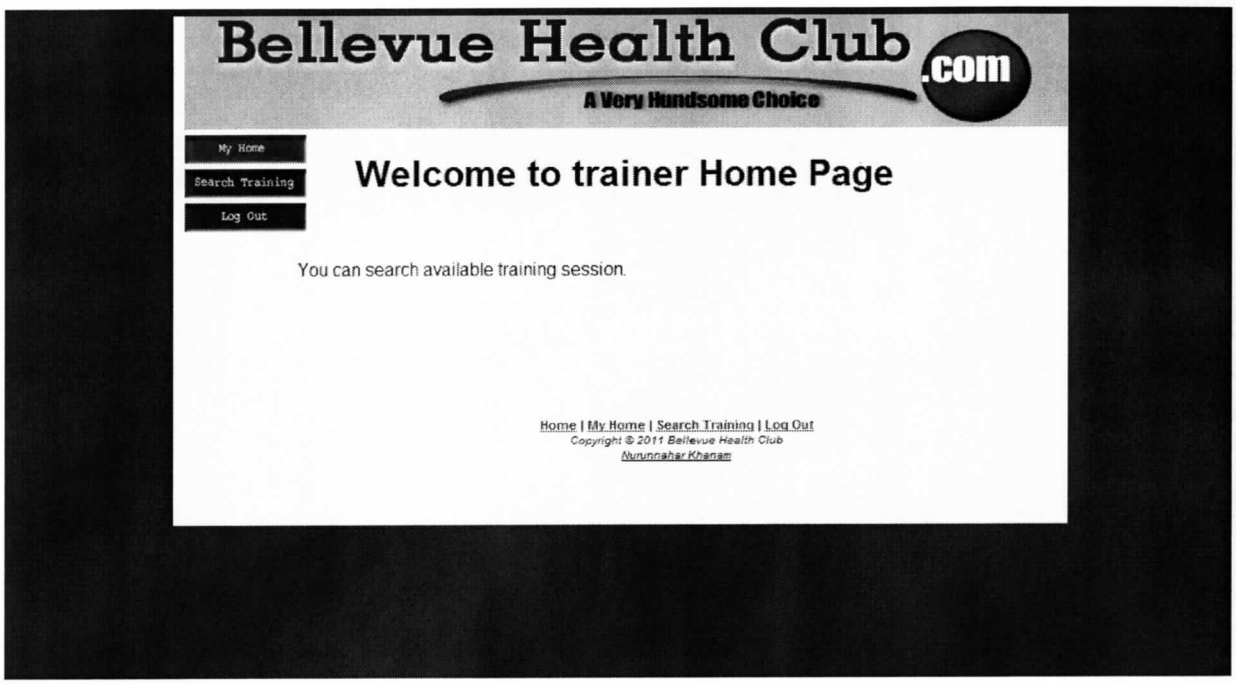


Figure 5-10 Trainer Home Page

Search Training Page: Trainer can go to search training page by clicking the "Search Training" button. Then trainer need to click "Search" button after selecting training place and insert training date and time.

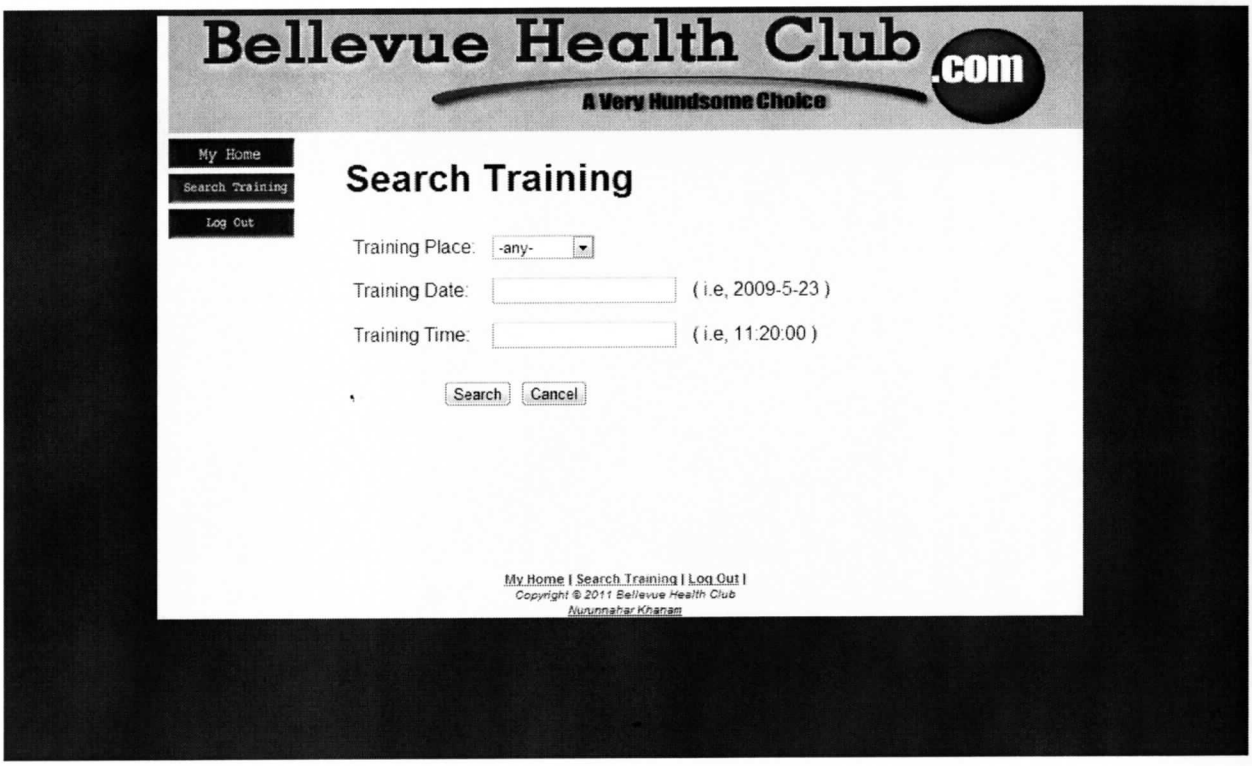


Figure 5-11Search Training Page

Search Result:

Trainer can see the search result here.

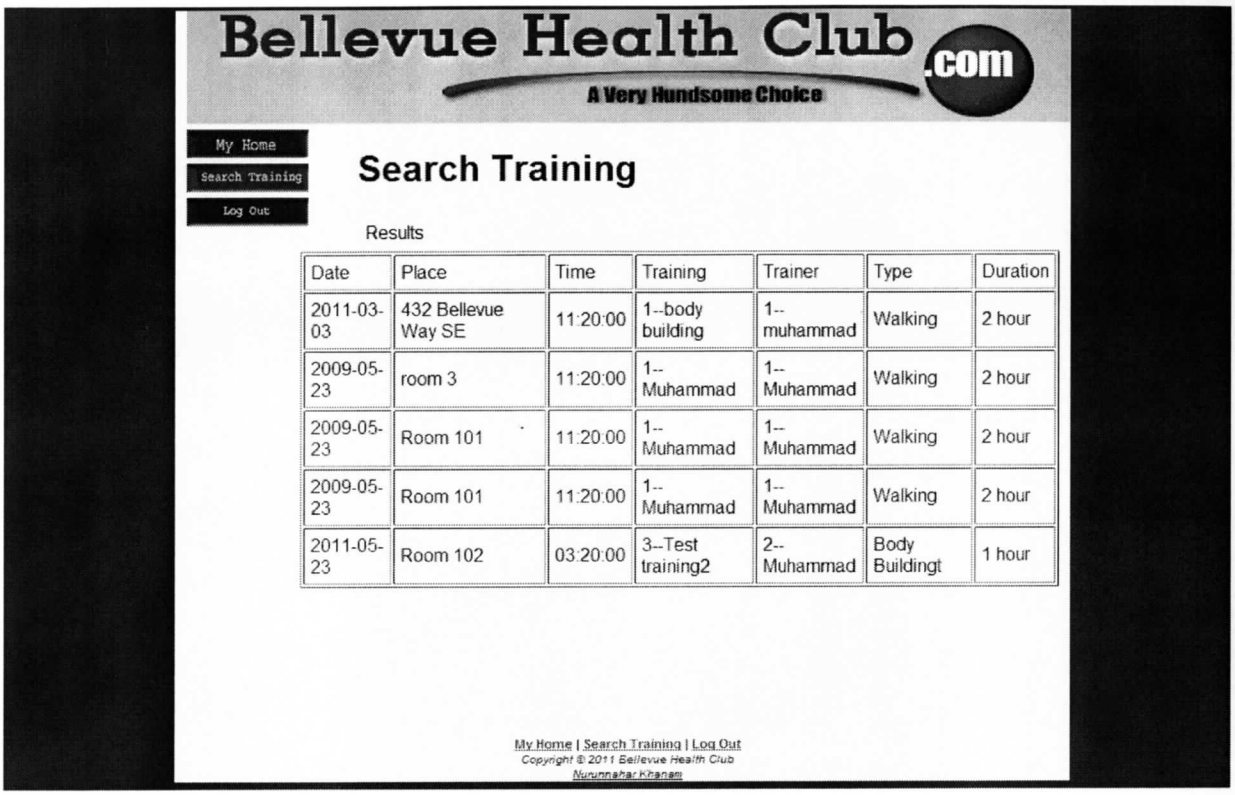


Figure 5-12 Search Result Page

Admin Panel:

Admin panel is for the admin of the system. Admin can login to the system from this panel to add user, trainer, and training, schedule training, and generate report.

Login Page:

Admin can login to the system from this page.

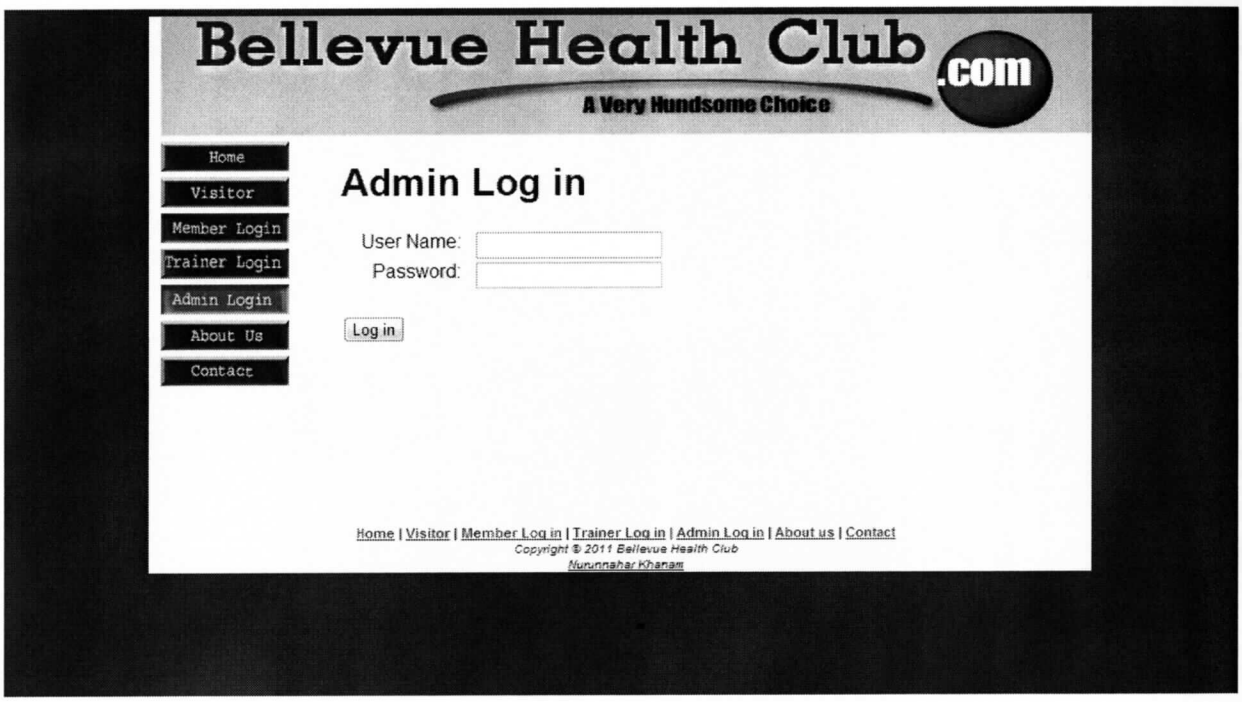


Figure 5-13 Admin Login Page

Admin Home Page: Admin can go to login page by clicking the "Admin Login" button of system home page.

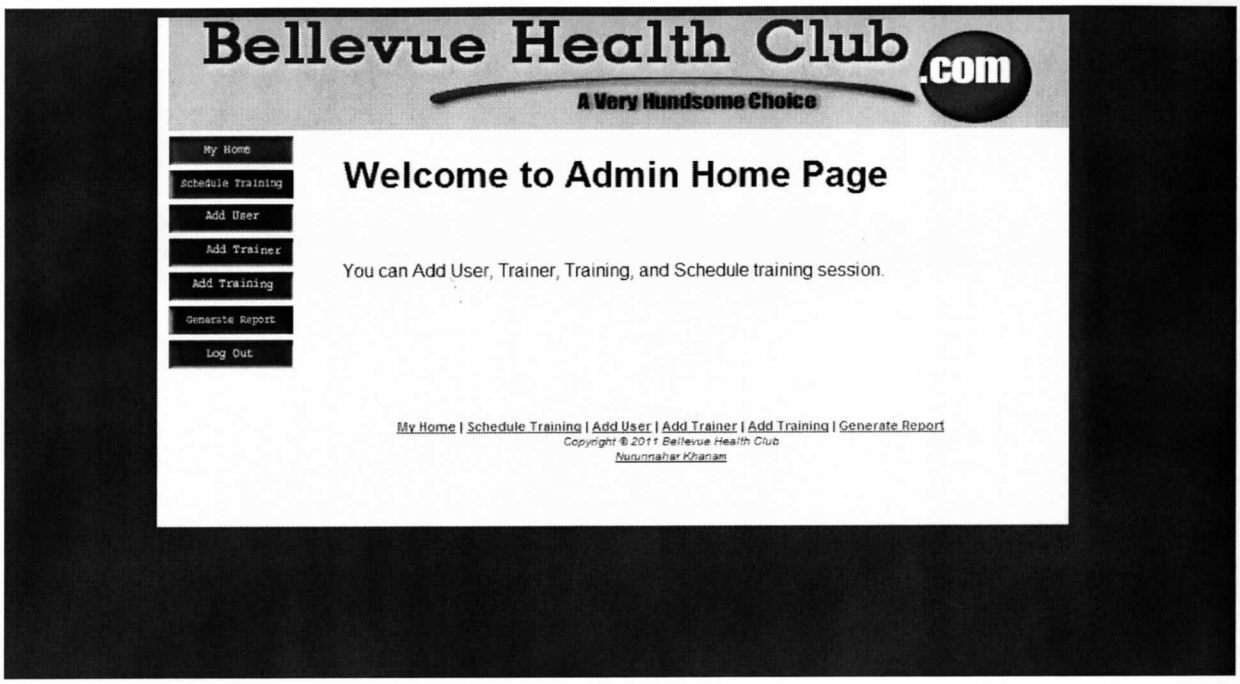


Figure 5-14 Admin Home Page

Add User Page:

Admin can add user from this page.

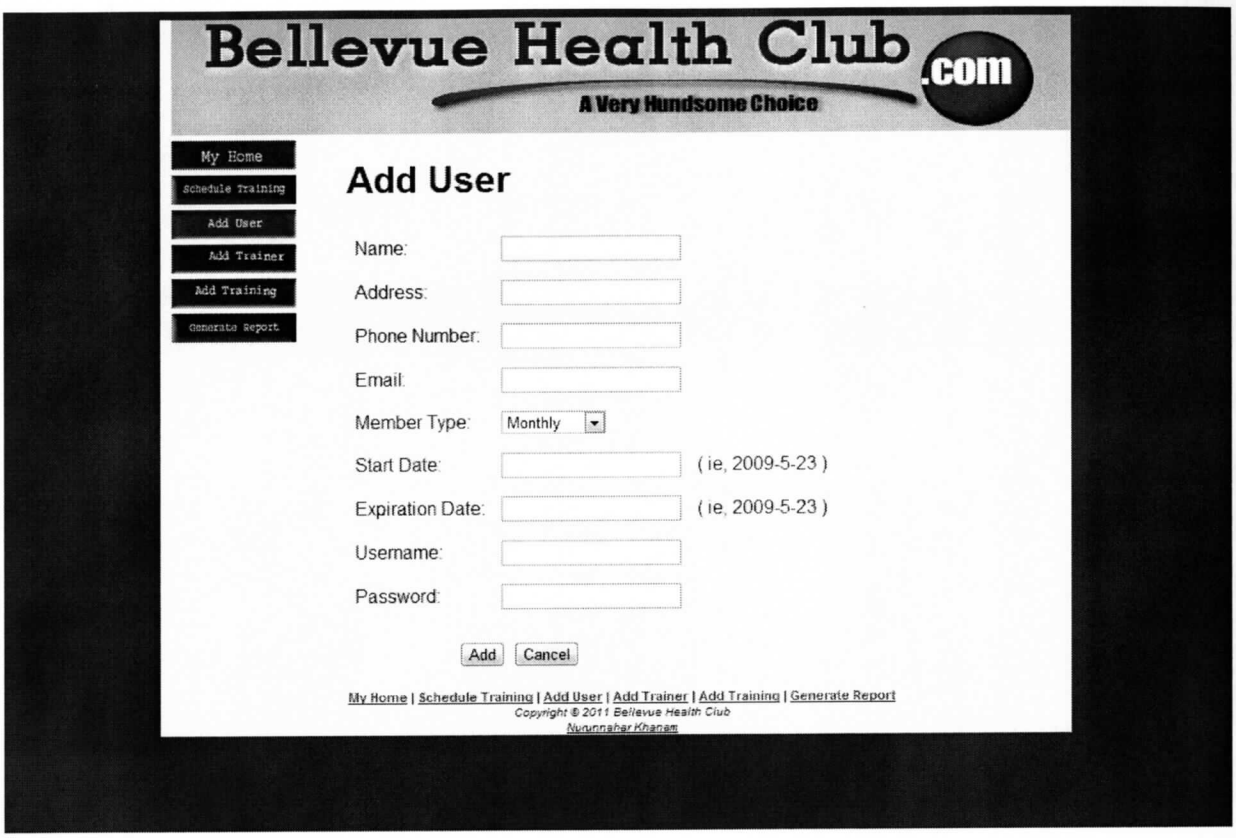


Figure 5-15 Add User Page

Add Trainer Page:

Admin can add trainer from this page.

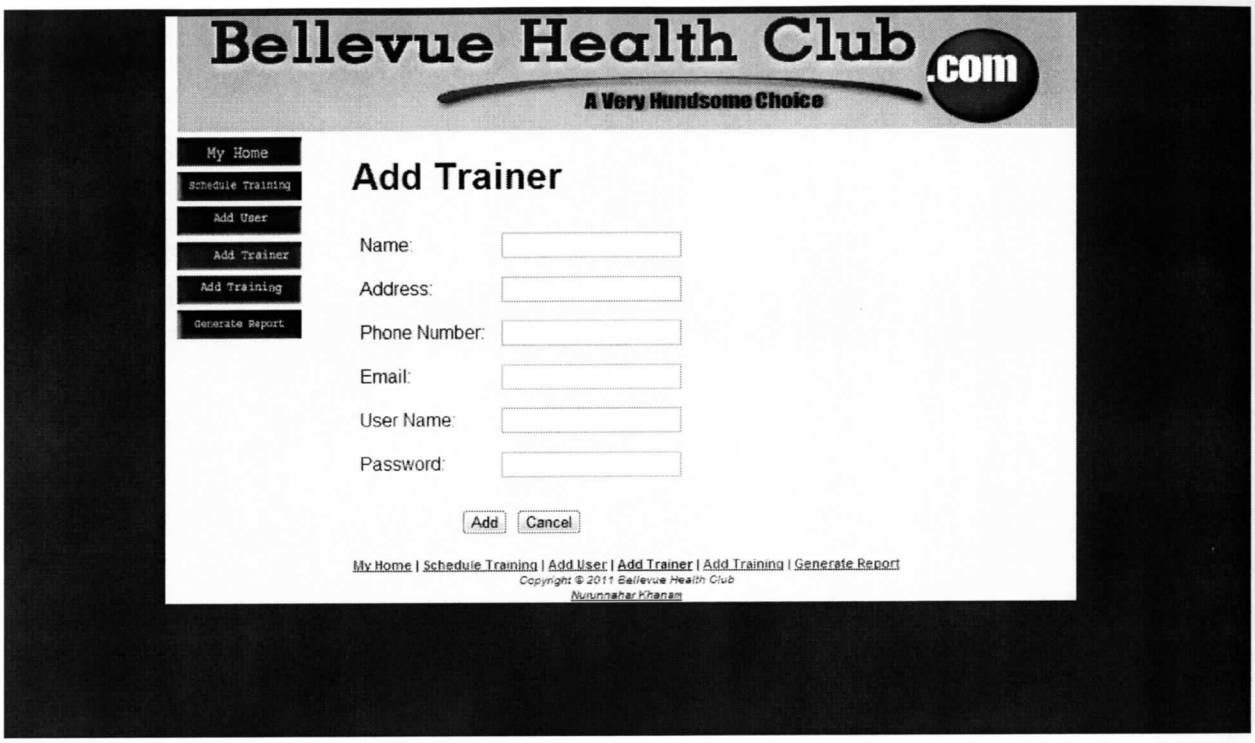


Figure 5-16 Add Trainer Page

Add Training Page:

Admin can add training from this page.

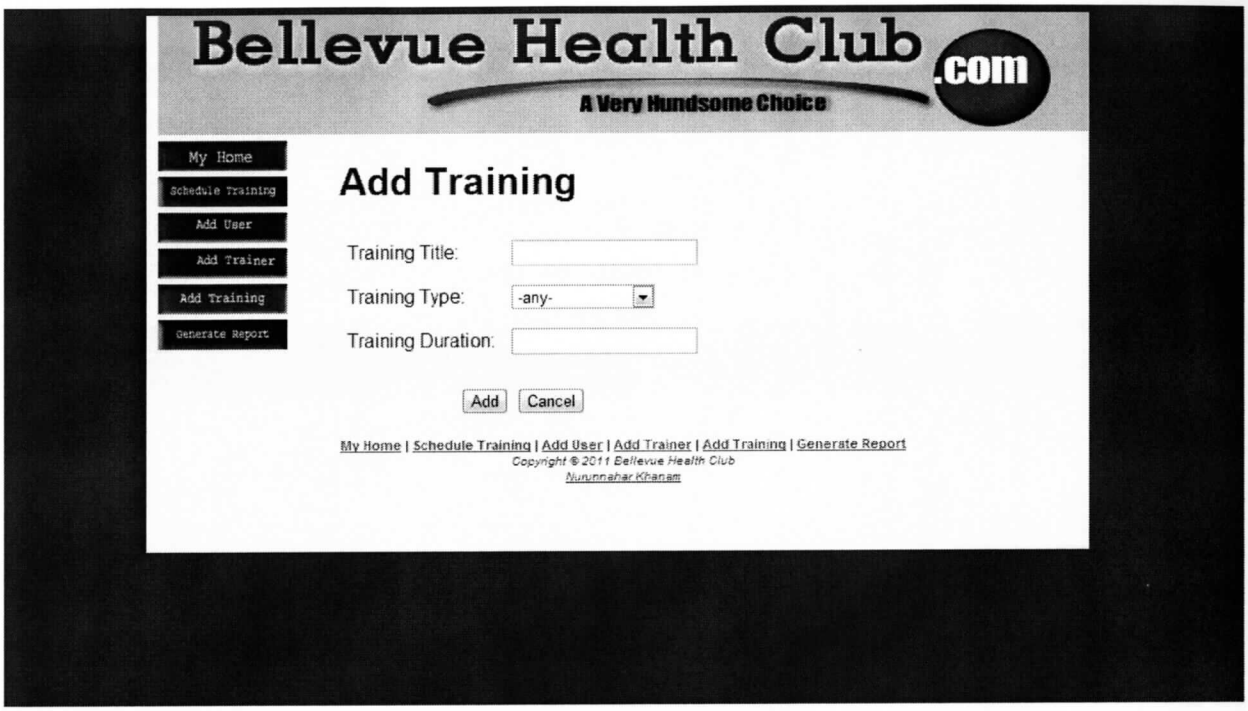


Figure 5-17 Add Training Page

Schedule Training Page:

Admin can schedule training from this page.

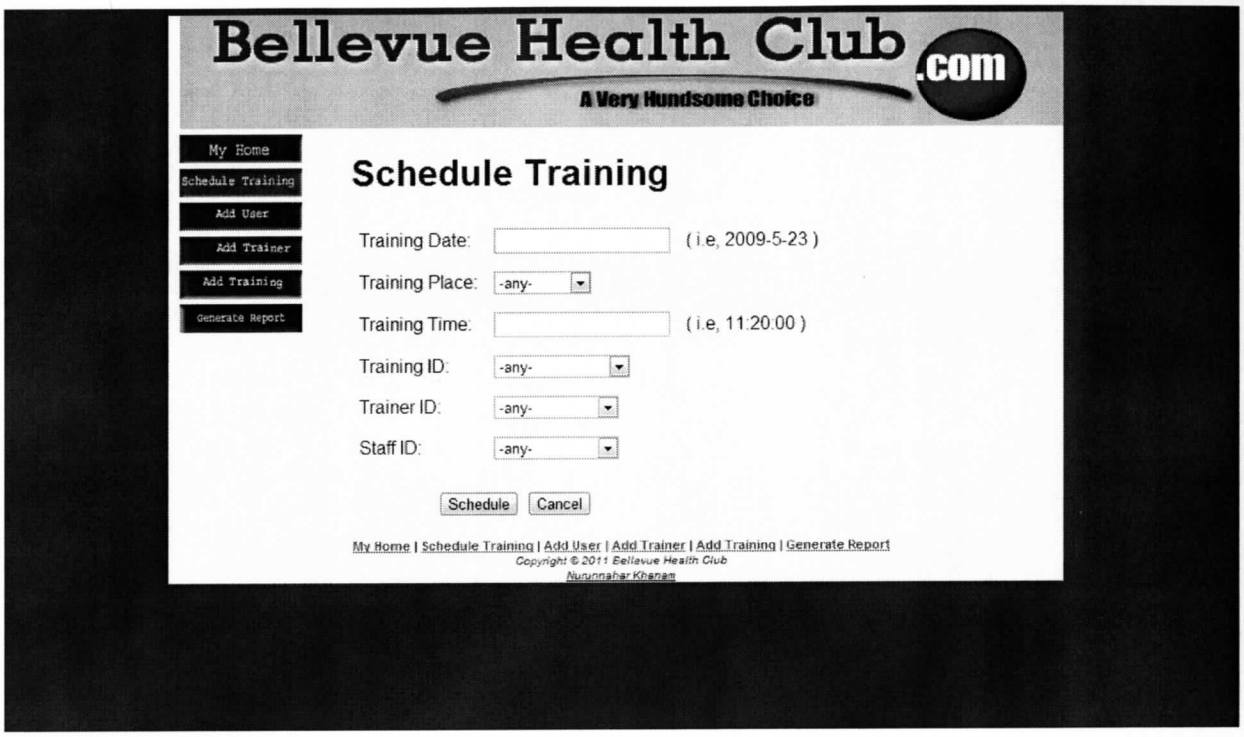


Figure 5-18 Schedule Training Page

Generate Report Page:

Admin can generate report from this page.

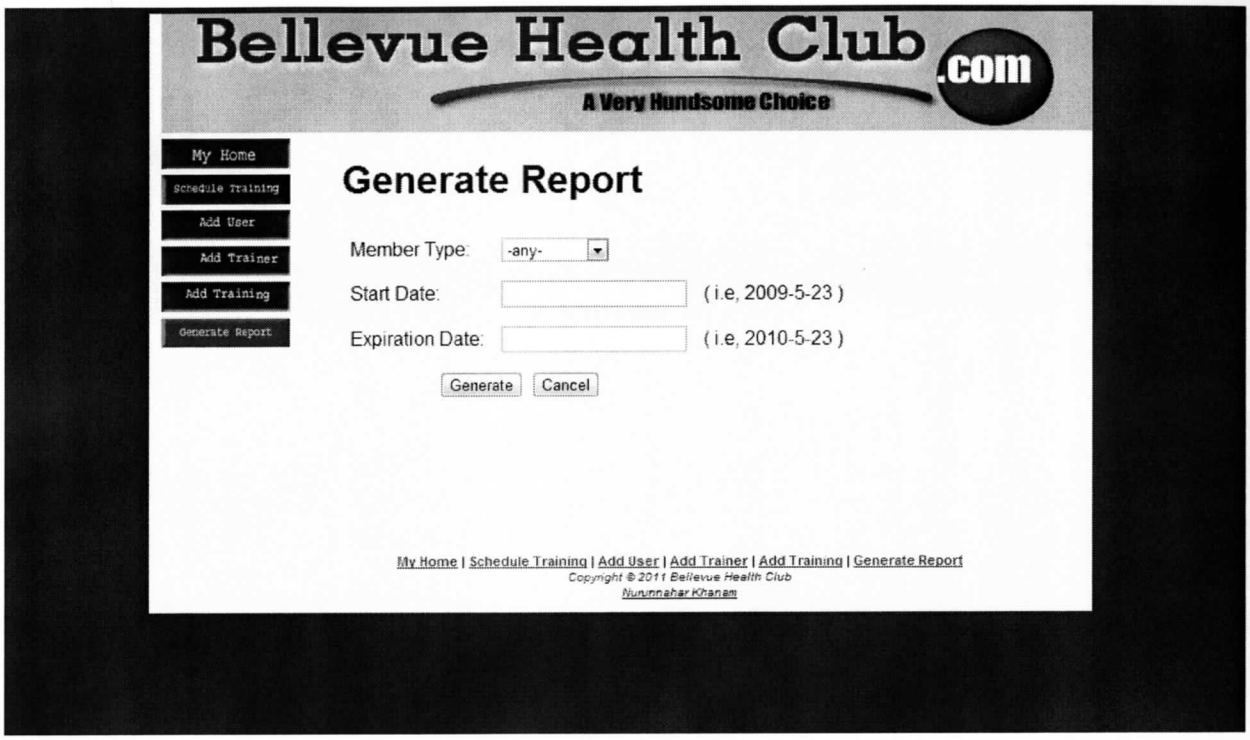


Figure 5-19 Generate Report Page

Generate Report Result Page:

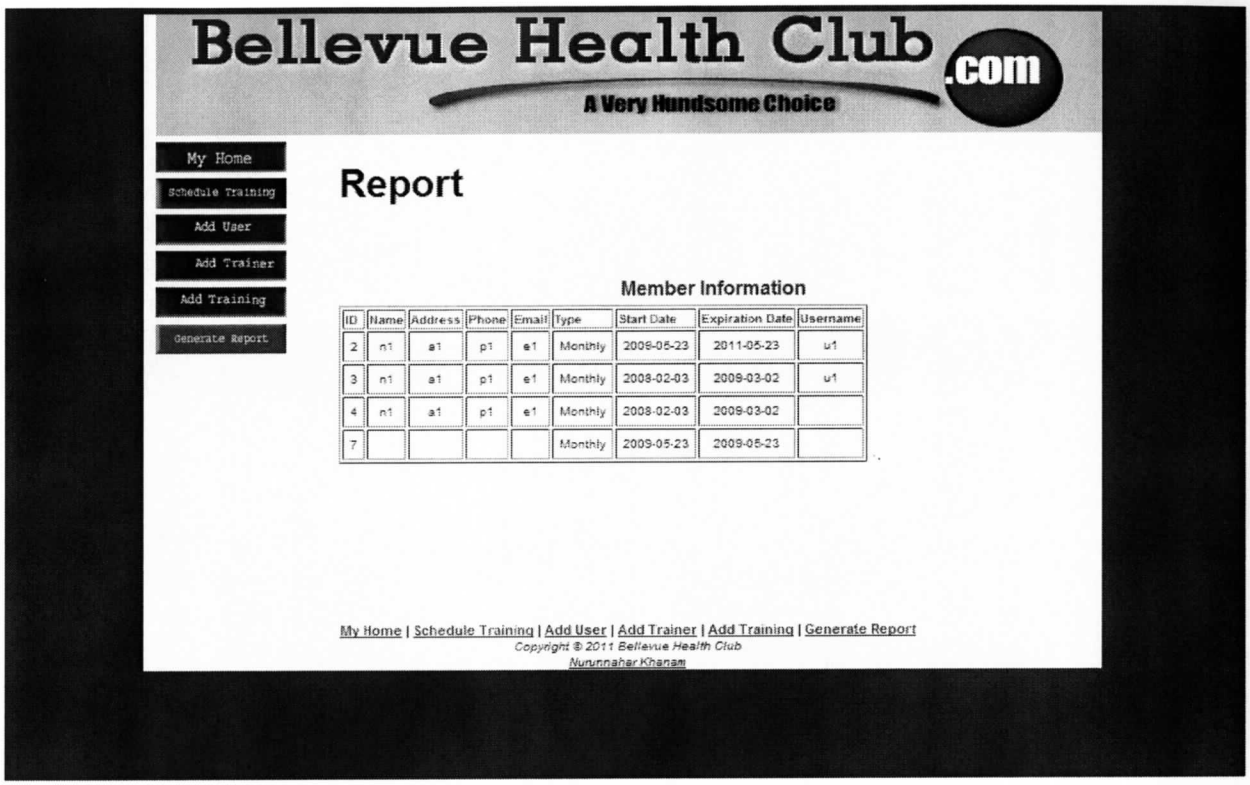


Figure 5-20 Report Result Page

CHAPTER 6

CONCLUSION

The project has achieved the objectives by using various system development methodologies. New system is completed with great success. Web based Interface module is being tested by members. Members are expressing extreme satisfaction with the services. Members have reported of time savings in using the new system. Admin is also satisfied with the new automated system. It makes their day-to-day operation easy. Members are also extremely satisfied and reported that the data entry is less error prone and more reliable. Members who tested the Web based interface have also expressed extreme satisfaction with the ability to view their account and other information about training session and bill. Even though the application meets all the initial specifications there are some areas where the BHC management department would like to expand the application. There is still an opportunity to improve the quality of the system and adding more features and functionalities to this application. The key areas where the application can be improved are

- Integrate payment process into the new System.

- Provide members the functionality to Report a Problem directly to the management.
- Integrate e-mail process to automatically send a report to the members into the new system.

Initiatives are well underway to expand the application to meet all the above goals.

CHAPTER 7

REFERENCES

http://www.w3schools.com/

http://www.go4expert.com/forums/showthread.php?t=421

http://www.quackit.com/html/html_table_generator.cfm

http://en.wikipedia.org/wiki/Multitier_architecture

Appendix A: Program Code

The appendix presents the details of the programming language code used in HCMS. The following code mainly consists of 6 languages, JSP, Servlet, java script, HTML, and MySQL, CSS. The code is divided into various .JSP, .Java, files; which perform different functions. Each JSP, .Java page carries a different function and will be processed by web server and presented to the users in plain HTML.

ATTACHED